

Université de Nice Sophia-Antipolis
Mémoire de stage
DEA Réseaux et Systèmes Distribués

Présenté par :
Abdelbasset TRAD

Sujet :
**Étude des mécanismes de contrôle
de la transmission de flux audio sur l'Internet**

Sous la direction de :
Mr Hossam AFIFI

Laboratoire d'accueil :
INRIA Sophia-Antipolis, Projet Planète



Soutenu le 1^{er} Juillet 2002

Remerciements

Je tiens à remercier Monsieur Hossam AFIFI pour son encadrement, ses conseils judicieux et ses remarques pertinentes qui m'ont aidé à mener à bien mon stage.

J'exprime ma profonde gratitude à Monsieur Walid DABBOUS pour la chance qu'il m'a accordé de faire mon stage de DEA au sein du projet Planète à l'Unité de Recherche INRIA Sophia-Antipolis.

Mes sincères remerciements vont à Monsieur Philippe NAIN, responsable du DEA Réseaux et Systèmes Distribués de l'Université de Nice Sophia-Antipolis.

Je remercie également les enseignants du DEA, pour la qualité de leurs enseignements et la disponibilité dont ils ont fait preuve à notre égard.

Table des matières

1	Introduction générale	7
2	Codage et transmission de l'audio sur l'Internet	8
2.1	Introduction	8
2.2	Codage audio	8
2.2.1	Principe du codage	8
2.2.2	Le codage différentiel (DPCM, ADPCM)	9
2.2.3	Le codage par synthèse (LPC, CELP, GSM)	9
2.2.3.1	Codage LPC (<i>Linear Predictive Coding</i>)	9
2.2.3.2	Codage CELP (<i>Code Excited Linear Predictive</i>)	9
2.2.3.3	Codage GSM (<i>Global System for Mobile Communications</i>)	9
2.3	Transmission de l'audio sur l'Internet	10
2.3.1	Suppression de silence	10
2.3.2	Paquetisation de l'audio numérisée	10
2.3.3	Transmission des données	10
2.3.4	Réception et décodage	11
2.4	Évaluation de quelques codecs normalisés	11
2.4.1	Le codec G.711	11
2.4.2	Les codecs G.729 et G.723.1	11
2.4.3	Le compromis taille/délai/débit	12
2.5	Conclusion	13
3	Problématiques de la transmission audio sur l'Internet	14
3.1	Introduction	14
3.2	Les contraintes de temps : le délai et la gigue	14
3.2.1	La problématique du délai	14
3.2.2	La gigue	16
3.3	La perte de paquets	16
3.4	Caractéristiques générales du délai et des pertes	16
3.4.1	Corrélation entre délai et pertes	16
3.4.2	Dépendance temporelle des pertes	17
3.5	Conclusion	17
4	Contrôle de la transmission audio sur l'Internet	18
4.1	Introduction	18
4.2	Mécanismes de contrôle de la gigue	18

4.3	Mécanismes de résolution des pertes	20
4.3.1	Schémas de la FEC (<i>Forward Error Correction</i>)	21
4.3.2	Mécanismes de contrôle à boucle fermée	22
4.4	Conclusion	23
5	Expérimentations	24
5.1	Introduction	24
5.2	Environnement des tests	24
5.2.1	Le réseau VTHD	24
5.2.2	La connexion de bout en bout testée	25
5.2.3	Outils utilisés pour les tests	25
5.3	Expérimentations et analyse des résultats	26
5.3.1	Mesure du débit des différents codecs	26
5.3.2	Débit d'une transmission de plusieurs flux audio	28
5.3.3	Délais de transmission audio sur le réseau haut débit	31
5.4	Conclusion	33
6	Conclusion et perspectives	34
A	Format des paquets RTP/RTCP	35

Table des figures

2.1	Schéma général d'une transmission audio sur l'Internet	10
3.1	Délai de l'émetteur : codage et mise en paquets	15
3.2	Modèle de Gilbert	17
4.1	Schéma général du contrôle de la gigue	19
4.2	Délai de compensation de la gigue	19
4.3	Schéma simple du mécanisme de FEC :	21
4.4	Schéma de transmission audio avec contrôle d'erreur adaptatif	21
4.5	Schéma de transmission d'une rafale audio en présence de perte	22
5.1	Réseau VTHD testé	24
5.2	Connexion testée : synchronisation des horloges des deux machines	25
5.3	Débit d'une transmission audio avec différents codecs	27
5.4	Débit pour 2 sources audio	28
5.5	Débit pour 5 sources audio	29
5.6	Débit pour 10 sources audio	30
5.7	Délai réseau d'un flux d'une transmission de 2 sources audio	32
A.1	Format de l'entête RTP	35
A.2	Format d'un paquet RTCP SR	37
A.3	Format d'un paquet RTCP RR	37

Liste des tableaux

2.1	Caractéristiques de quelques codecs normalisés	12
2.2	Taille des entêtes IP/UDP/RTP	13
2.3	Comparaison des paramètres des codecs normalisés	13
5.1	Codecs utilisés pour les tests	26
5.2	Débit réel des codecs audio	26
5.3	Débit pour une transmission de plusieurs flux audio	28
5.4	Effet du délai de bout en bout sur la qualité de l’audio	32

Chapitre 1

Introduction générale

Le réseau Internet a été conçu à l'origine pour transporter les données. Plusieurs protocoles et applications ont été développés afin de mettre en évidence les avantages de l'interconnexion d'un grand nombre d'ordinateurs répartis dans des sites géographiquement distants. Les applications de transfert de fichiers, du courrier électronique et d'accès à des machines distantes constituent encore maintenant une large fraction du trafic véhiculé sur les réseaux de données. Ces applications n'ont pas de contraintes temporelles fortes, elles ne sont pas des applications temps réel.

L'évolution technologique constante a conduit au développement et l'utilisation de machines de plus en plus puissantes et à l'augmentation du débit des liaisons réseaux, ce qui a fait surgir l'intérêt d'utiliser l'Internet pour envoyer de l'audio et de la vidéo, types de données qui traditionnellement étaient réservés aux réseaux spécialisés à cet effet, et depuis déjà quelques années l'audio et la vidéoconférence sont devenus une pratique courante. Mais la nature même de l'Internet, fait que ce réseau ne soit pas adapté pour la transmission des données temps réel. Par exemple, une conversation audio à travers le réseau nécessite que le délai de traversée du réseau ne dépasse pas quelques centaines de millisecondes. Ceci a comme conséquence que la qualité de l'audio envoyé à travers l'Internet a en moyenne une qualité médiocre.

Plusieurs approches et mécanismes ont été proposés pour permettre à une application d'audioconférence ou de téléphone sur Internet, de maintenir une qualité auditive acceptable même dans des cas où le réseau est assez congestionné. Ce stage a pour objectif d'étudier et de tester la performance des mécanismes de la transmission de flux audio sur l'Internet.

Le rapport est organisé de la manière suivante : dans le Chapitre 2 on présente une étude comparative des différents mécanismes de codage audio ainsi que des différents codecs normalisés [Codecs]. Le Chapitre 3 s'adresse précisément à l'analyse des problèmes liés à la transmission d'information audio sur l'Internet, on présente les différentes approches proposées pour les résoudre au Chapitre 4. Ces solutions, sous la forme de mécanismes de contrôle, sont implémentées sur le logiciel d'audioconférence et téléphone sur Internet RAT (*Robust Audio Tool*) [RAT] que nous avons utilisé pour effectuer des expérimentations décrites au niveau du Chapitre 5. Une analyse des résultats de mesures effectuées ainsi qu'une étude du comportement qu'auraient ces mécanismes dans un Internet évoluant vers le haut débit sera abordée aussi dans ce chapitre. Enfin, nous terminerons par une conclusion récapitulative et les perspectives des travaux futurs.

Chapitre 2

Codage et transmission de l'audio sur l'Internet

2.1 Introduction

Les applications audio sur un réseau font intervenir deux aspects distincts qui sont d'une part la numérisation et le codage des données audio, et d'autre part la mise en forme (paquetisation) de ces données pour être transmises sur le réseau.

Dans ce chapitre on décrit brièvement les grands types de codage audio, les différentes phases d'une transmission audio sur Internet et on présente une étude comparative de quelques codecs normalisés.

2.2 Codage audio

Le but du codage est de transformer un signal vocal ou musical en général analogique, en un signal numérique d'un débit et d'une qualité donnés.

2.2.1 Principe du codage

La première opération du codage est l'échantillonnage du signal analogique à une certaine fréquence et une certaine précision, cette précision étant caractérisée par le nombre de bits utilisés pour coder l'amplitude de chaque échantillon. Le choix de la fréquence et du nombre de bits utilisés répond à un compromis débit/qualité du signal codé. Plus grande est la qualité souhaitée, plus important est le débit obtenu après échantillonnage.

L'algorithme de codage le plus simple est celui qui revient seulement à échantillonner un signal analogique et à quantifier les échantillons (les convertir des valeurs réelles en valeurs de précision finie). Ce codage appelé PCM (*Pulse Code Modulation*) [ITU88], est utilisé pour coder la voix dans le réseau téléphonique. La bande passante d'une paire torsadée étant de 3,5 KHz, la fréquence d'échantillonnage est fixée à 8 KHz¹ (un échantillon toutes les 125 μ s). La quantification est faite avec une échelle logarithmique sur 8 bits, ce qui implique un débit d'émission de 64 kbit/s.

Le codage PCM est à la base d'une famille de codages différentiels largement utilisés, et qui seront décrits dans le paragraphe suivant.

¹Respectant la règle de Shanon

2.2.2 Le codage différentiel (DPCM, ADPCM)

Les échantillons successifs d'une source audio sont fortement corrélés. Il semble donc judicieux d'encoder non pas les échantillons eux même mais la différence entre des échantillons successifs. Partant de ce principe, le codage DPCM (*Differential PCM*) consiste à prédire l'échantillon $n+1$ à partir des échantillons n , et également $n-1$, $n-2$, etc. Il est donc possible de n'encoder que la différence entre un échantillon x_n et sa valeur prédite $\hat{x}_n$. Dans le cas du codage DPCM, la prédiction de x_n est simplement la valeur de l'échantillon précédent x_{n-1} . Dans un cas plus général, une prédiction linéaire peut être utilisée. On obtient alors un codage de type ADPCM (*Adaptive Differential PCM*) [ITU90] qui réduit le nombre de bits par échantillon à 2-5 bits.

Les codages différentiels ne nécessitent que quelques opérations algébriques, ils sont peu coûteux de point de vue temps de calcul CPU.

2.2.3 Le codage par synthèse (LPC, CELP, GSM)

Le codage par synthèse est très différent du codage différentiel. L'idée n'est plus de manipuler les échantillons pour éliminer la redondance et les corrélations entre échantillons, mais au contraire de considérer des blocs d'échantillons et de construire un modèle qui produise des échantillons statistiquement identiques (ou proches) des échantillons originaux. Ce modèle permet de synthétiser des échantillons avec des propriétés statistiques données.

2.2.3.1 Codage LPC (*Linear Predictive Coding*)

Le codage LPC consiste à synthétiser des échantillons à partir d'un modèle d'un système de production vocale et d'une excitation (sinusoïde ou bruit blanc) [Gar96].

2.2.3.2 Codage CELP (*Code Excited Linear Predictive*)

C'est une extension du codage LPC. Il comporte toujours deux phases, correspondant aux fonctions d'excitation et de transfert [Gar96]. Par contre, la fonction d'excitation n'est pas seulement un bruit blanc ou un sinusoïde, mais une combinaison linéaire de fonctions stochastiques (c'est à dire de bruit) et périodiques.

2.2.3.3 Codage GSM (*Global System for Mobile Communications*)

Le codage GSM [Sem99] utilise une variante du codeur CELP appelée RPE (*Regular Pulse Excited*), la fonction d'excitation fait intervenir des impulsions périodiques couplées à une boucle de prédiction à long terme. Le codage obtenu est de qualité très proche de PCM, mais avec un débit de 13 kb/s seulement.

À l'opposé des codages différentiels, les codages par synthèse n'effectuent pas moins de 3 calculs d'autocorrélation et 2 calculs algorithmiques pour chaque fenêtre de 20 ms. Des coûts CPU trop élevés rendent l'utilisation de codeurs de bas débit de type LPC pénalisante pour des machines peu puissantes de type PC.

2.3 Transmission de l'audio sur l'Internet

Après avoir décrit le codage de données audio, nous considérons maintenant la transmission de ces données codées. Pour cela, on considère le schéma suivant (Figure 2.1) décrivant la structure globale d'un outil audio pour l'Internet. Les phases importantes seront décrites ci-dessous.

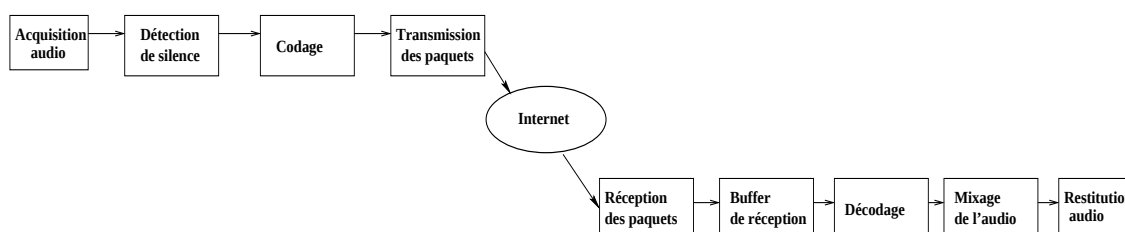


FIG. 2.1 – Schéma général d'une transmission audio sur l'Internet

2.3.1 Suppression de silence

La voix est formée d'une alternance entre des périodes d'activité (*talkspurts*) et des périodes de silence, une technique visant à réduire la bande passante réseau utilisée par une source audio consiste à ne pas transmettre pendant les périodes de silence. Les périodes d'activité d'une source de voix constituant en général 40% seulement du temps total [Bra65], la bande passante consommée peut être ainsi réduite de manière significative.

2.3.2 Paquetisation de l'audio numérisée

Les échantillons audio codés par la source sont collectés et stockés dans un buffer. Après une période de temps fixe : intervalle de paquetisation, ces échantillons sont mis dans un paquet destiné à être transmis sur le réseau. Les intervalles de paquetisation typiques sont de 10-50 ms [Sch92]. L'intervalle de paquetisation étant fixé, le schéma de codage détermine le nombre de bits par paquet et par conséquent le débit de la transmission audio.

2.3.3 Transmission des données

La transmission d'un flux audio sur l'Internet peut se faire en utilisant l'un des deux protocoles de transport disponibles, à savoir TCP [RFC793], ou UDP [RFC768]. Le protocole TCP a comme avantage qu'il chemine des données de façon fiable : sans pertes ni déséquencement. Cependant, TCP obtient cette fiabilité par l'utilisation d'acquittements, la retransmission de paquets perdus et le contrôle de flux par fenêtre de congestion. Ces mécanismes sont incompatibles avec les contraintes temps réel associées aux applications audio.

L'autre solution consiste donc à utiliser un protocole de transport minimal et à implémenter dans l'application les fonctionnalités spécifiques à la transmission audio. Le protocole UDP ne fournit pas de fiabilité, mais il ne comporte pas de mécanismes comme la retransmission qui augmentent les délais entre sources et destinations.

Le protocole RTP (*Real-time Transport Protocol*) [RFC1890][HAH95] est utilisé conjointement avec UDP afin d'assurer une fiabilisation de bout en bout. RTP comprend deux parties, une partie “données” et une partie “contrôle”. Principalement RTP permet de :

1. identifier le type de l'information transportée (par exemple audio LPC, ou audio PCM ou vidéo),
2. ajouter des estampilles de temps (*timestamps*) pour indiquer l'instant auquel les données ont été générées.
3. détecter les paquets perdus à la destination en utilisant un numéro de séquence.

La partie “contrôle” de RTP, à savoir le protocole RTCP (*Real-time Transport Protocol*) [RFC1889] permet de fournir à l'émetteur un “*feedback*” sur la qualité de la transmission (taux de perte, gigue), mais il n'offre pas une garantie de qualité de service. Une description plus précise du format des paquets RTP et RTCP est donnée en Annexe A.

2.3.4 Réception et décodage

À la réception des paquets disponibles sur le réseau, l'information audio est extraite de ces paquets et placée dans un buffer de réception. L'un des schémas de décompression disponibles est utilisé pour restituer l'information audio.

2.4 Évaluation de quelques codecs normalisés

Les codecs les plus utilisés en téléphonie sur IP sont : le G.711 [ITU88], le G.729 [ITU96b] et le G.723.1 (6,3 kbps) [ITU96a]. Tous ces codecs sont basés sur des mécanismes conçus à l'origine pour la téléphonie à commutation de circuits, en particulier la téléphonie mobile.

Les codecs G.729 et G.723.1 [Cox97] sont basés sur le codage CELP (décrit en Section 2.2.3), ils ne réalisent pas de bonnes performances de fonctionnement dans le cas des réseaux à commutation de paquets vu les mécanismes de contrôle d'erreurs utilisés qui sont orientés bits et ne sont pas appropriés à la résolution de la perte de paquets.

2.4.1 Le codec G.711

Les points les plus importants concernant le codec G.711 sont les suivants :

- Basé sur un codage différentiel PCM, ce codec ne présente aucun mécanisme de contrôle des pertes de paquets, la qualité de la voix transmise se dégrade remarquablement avec l'augmentation du nombre de paquets perdus.
- Des mécanismes de contrôle d'erreur basés sur l'extrapolation/interpolation des segments de voix reçus ont été introduits afin d'améliorer la qualité de l'audio reconstituée (G.711 Annexe). Cependant ces mécanismes se basent sur des heuristiques et ne garantissent pas la robustesse de la transmission audio sur le réseau.

2.4.2 Les codecs G.729 et G.723.1

Les codecs G.729 et G.723.1 appartiennent à une classe de codage différente de celle du G.711 qui est le codage par synthèse CELP. On peut résumer leurs principales caractéristiques par les points suivants :

- La qualité audio de ces codecs est moins bonne que celle du PSTN ²(*Public Switched Telephone Network*), ils ont une qualité comparable à celle de la téléphonie mobile.
- La perte de paquets cause une grande dégradation de la performance due à la propagation de l'erreur liée à la nature du codage qui se base sur la dépendance entre les paquets.
- Des mécanismes basés sur des méthodes heuristiques de contrôle d'erreur ont été ajoutés à ces codecs, ils sont affectés eux aussi par la dépendance entre les paquets codés (plusieurs paquets sont nécessaires pour remédier à la perte d'un seul).
- Ces codecs ne sont pas flexibles vu la taille fixe des paquets.
- Ils présentent une complexité d'implémentation accrue.

Dans le Tableau 2.1 on résume quelques caractéristiques de performance des codecs G.711, G.729, G.723.1 et GSM.

Codec	Type de codage	Débit (kbps)	Robustesse aux pertes	Complexité	Coût DSP
G.711	Numérisation PCM 64 kbps	64	Faible	Faible	Faible
G.729	Différentiel prédictif	8	Très faible	Élevée	30 MIPS
G.723.1	Différentiel prédictif	6,3/5,3	Très faible	Élevée	17 MIPS
GSM	Codage par synthèse	13	Très faible	Élevée	15 MIPS

TAB. 2.1 – Caractéristiques de quelques codecs normalisés

Le Tableau 2.1 montre le compromis entre le débit, la complexité et le coût en traitement des codecs : les deux codecs G.723.1 et G.729 présentent des caractéristiques intéressantes pour la transmission audio sur l'Internet (bas débits), leur complexité élevée ne présente pas une vraie contrainte vu la croissance rapide des puissances CPU. Cependant leur robustesse aux pertes est faible vu qu'ils ne présentent pas de mécanismes de contrôle d'erreur de la transmission audio sur des réseaux à commutation de paquets.

2.4.3 Le compromis taille/délai/débit

Les codecs de téléphonie sur IP sont soumis à deux contraintes principales :

- Les ressources en bande passante.
- Le délai de transmission sur l'Internet qui ne doit pas dépasser 200 ms.

De ces deux contraintes, la première va disparaître grâce à l'abondance de la bande passante offerte par les liens de transmission optique et les technologies WDM³. Il faut donc réduire les délais. Pour réduire le délai initial et le débit de sortie, le codeur dispose de deux moyens d'action : le taux de compression et la longueur des paquets.

Le taux de compression

Une baisse substantielle du débit de codage se traduit par une diminution marginale du débit réel vu le poids du trafic d'en-têtes (voir Tableau 2.2), sauf si des techniques de compression d'en-têtes [RFC2508] sont utilisées.

²Réseau Téléphonique public Commuté (RTC)

³Multiplexage en longueur d'onde

Protocole	IP v4/v6	UDP	RTP
Taille de l'entête (octets)	20/40	8	12

TAB. 2.2 – Taille des entêtes IP/UDP/RTP

La longueur des paquets

Le paquet de voix IP doit être émis toutes les 20, 30 ou 40 ms afin de ne pas introduire davantage de latence dans la transmission. À un débit audio constant de 8kbps, un paquet de 40 ms muni des en-têtes RTP/UDP/IP (Tableau 2.2) génèrera un débit réel de 16 kbps, alors qu'un paquet de 20 ms génèrera un débit réel de 24 kbps, d'où l'impact limité de la réduction du débit du codec sur le débit réel transmis. D'autre part, la longueur de la trame et du *look-ahead* (un prétraitement sur la trame suivante) doit rester suffisante pour avoir une bonne qualité de restitution [Sus00]. Le Tableau 2.2 ci-après [KBS98] fournit quelques valeurs de longueurs de trames, de *look-ahead*, de débit et de débit réel des codecs normalisés.

Codec	Trame + look-ahead	Débit (kbps)	Débit avec les entêtes IP/UDP/RTP (kbps)
G.711	10 ms + 5 ms	64	80
G.729	10 ms + 5 ms	8	24
G.723.1	30 ms + 7,5 ms	6,3/5,3	17/16
GSM	20 ms + 10 ms	13	29

TAB. 2.3 – Comparaison des paramètres des codecs normalisés

D'après le Tableau 2.3 on peut noter que les codecs G.711 et G.729 ont la même taille de trame mais le débit du G.729 est beaucoup moins élevé vu la technique de compression et la nature du codage différentiel prédictif utilisé. Le codec G.723.1 est caractérisé par un très bas débit mais par une longueur de trame et d'un *look-ahead* plus importants que ceux du G.729 : diminuer le débit nécessite plus d'information contenue dans la trame et plus de temps de traitement (*look-ahead*) sur cette trame afin de prédire la trame suivante. Un *look-ahead* encore plus important caractérise le codage GSM vu la boucle de prédiction à long terme utilisée. L'avantage du codage GSM consiste au fait qu'il assure une qualité très proche du codage G.711 avec des débits moins élevés.

2.5 Conclusion

Dans ce chapitre on a présenté brièvement les principes du codage et les mécanismes de la transmission audio sur l'Internet. En une seconde partie, on a établi une étude comparative de quelques codecs normalisés.

Il est à noter que la taille des paquets générés par les différents codecs audio décrits dans ce chapitre est fixe vu qu'ils ont été conçus à l'origine pour des réseaux à commutation de circuits où les ressources sont dédiées durant toute une communication. La cadence régulière de génération de paquets de taille constante n'est pas bien appropriée aux réseaux à commutation de paquets, i.e. l'Internet, caractérisés par des variations imprévisibles des paramètres réseau (bande passante disponible, taille des files d'attente des routeurs), ce qui cause certains problèmes pour la transmission audio sur l'Internet qu'on va analyser au chapitre suivant.

Chapitre 3

Problématiques de la transmission audio sur l'Internet

3.1 Introduction

Le réseau Internet, n'offrant qu'un service "*best effort*" caractérisé par des pertes de paquets très variables et ne garantissant pas des limitations de délais, affecte la qualité de la voix transmise sur ce réseau. Les principaux défauts de la transmission de l'audio sur IP auxquels on s'intéresse dans ce chapitre sont :

- Le délai de bout en bout qui doit rester inférieur à 400 ms (aller-retour), pour satisfaire les critères d'interactivité d'une communication téléphonique [Kle67][Bra71].
- La gigue qui représente la variation du délai. Au cas où le délai est constant, la synchronisation du signal entre l'émetteur et le récepteur est préservée. Dans le cas contraire, les variations de délais détruisent la base de temps du signal reçu, obligeant le destinataire à maintenir une mémoire tampon de resynchronisation (buffer de gigue).
- Les pertes de paquets : dans les moments de congestion extrême, le taux de pertes dépassant un certain seuil remet en cause la qualité et l'intégrité de la voix.

3.2 Les contraintes de temps : le délai et la gigue

3.2.1 La problématique du délai

Dans ce paragraphe, on examine les causes du délai. Le délai dépend de la technologie de compression : les algorithmes de codage (c.f. Section 2.2) manipulent des trames de taille comprise entre 10 et 30 ms, augmentée d'un "*look-ahead*" qui constitue un prétraitement sur la trame suivante.

En téléphonie sur IP, à l'opposé d'un réseau à commutation de circuits, le signal ne commence à être transporté vers sa destination qu'après un délai de remplissage du paquet. Le paquet rempli, est émis sur le réseau, ce qui introduit un délai de transmission supplémentaire engendré par les noeuds et routeurs du réseau. Le délai de bout en bout peut être décomposé en trois éléments contributeurs : l'émetteur, le réseau, et le destinataire.

Le délai de l'émetteur

1. La numérisation et le codage : La numérisation est génératrice de délais plus ou moins importants selon la puissance des machines. Le signal est codé en PCM.
2. La compression : Le délai de compression est constitué de trois parties. Le délai de trame (10 à 30 ms) et le *"look-ahead"* formant le "délai algorithmique", auxquels s'ajoute le temps de traitement des processeurs (Figure 3.1).
3. La mise en paquets : Un délai de paquets est mis par l'application pour effectuer des tâches liées à la constitution des paquets (établissement des en-têtes...), il est inférieur à 5 ms [Sus00]. Le temps du remplissage d'un paquet constitué d'une, deux ou trois trames sera de 40 à 50 ms, ce temps constitue un délai initial strictement incompressible. Il est alors tenant de construire des paquets ne contenant qu'une seule trame, mais la fréquence d'émission des paquets serait extrêmement élevée et dépasserait les possibilités des interfaces réseaux. Le trafic d'en-têtes généré en l'absence de techniques de compression d'en-têtes engorgerait le réseau avec des *overheads*.
4. La transmission : Le temps d'accès au réseau par l'interface peut représenter quelques dizaines de millisecondes, correspondant à la file d'attente de sortie et au temps d'accès au lien.

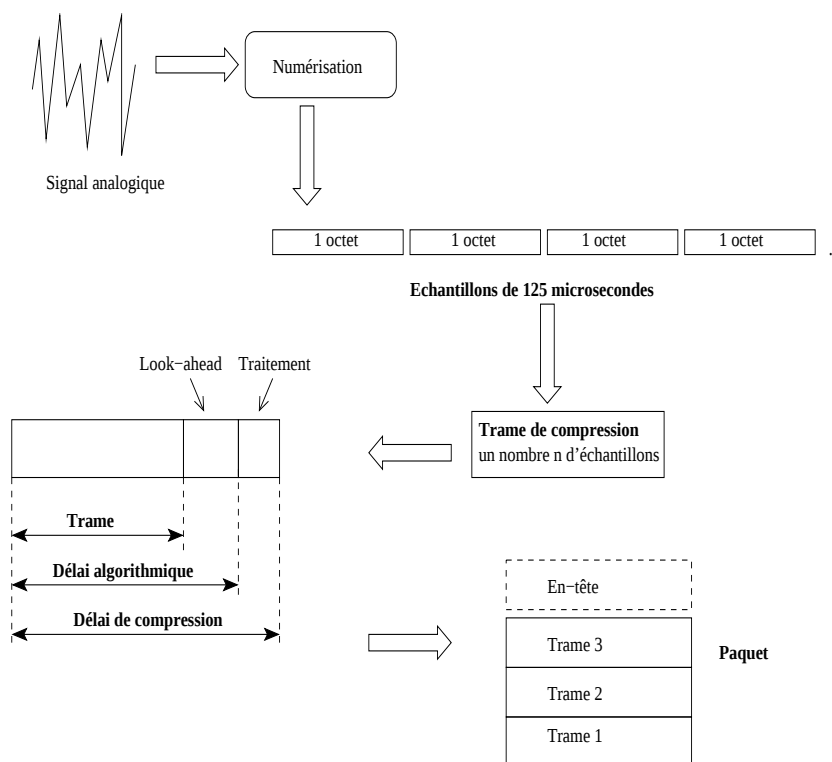


FIG. 3.1 – Délai de l'émetteur : codage et mise en paquets

Le délai réseau

Le délai réseau est constitué de :

- délai de propagation
- délai de transmission
- délai d'attente des paquets dans les files des routeurs.

Le délai de propagation est constant, il est de plus en plus faible grâce aux nouvelles technologies de transmission optique, $5\mu s/Km$ pour une fibre optique [Jia99]. Le délai de transmission est proportionnel à la taille des paquets. Le temps de la commutation est très faible (inférieur à 1 ms) [Sus00], par contre le passage des paquets dans les files d'attente des routeurs est très variable car dépendant du trafic instantané.

Le délai du récepteur

Il est composé du délai de réception sur l'interface réseau, du délai introduit par le buffer de gigue pour la resynchronisation, du temps nécessaire pour la dépaquetisation consistant à enlever les en-têtes et à remettre les trames compressées au décodeur.

3.2.2 La gigue

La gigue est causée par les délais d'attente des paquets aux routeurs du réseau. Les paquets envoyés par l'émetteur à des intervalles de temps équidistants arrivent à la destination à des intervalles non-équidistants causant des discontinuités (*gaps*) de la voix restituée. Les mécanismes de contrôle de la gigue seront examinés au Chapitre 4.

3.3 La perte de paquets

À l'opposé de la transmission de données, l'audio ne nécessite pas une fiabilité stricte de la transmission, mais sa tolérance à la perte de paquets est faible. Il est estimé que la voix restituée à la destination perd son intelligibilité si les pertes dépassent 2% [Jay81].

La perte de paquets est principalement causée par :

1. Une congestion des routeurs résultant en un rejet de paquets audio.
2. Des délais de transmission assez élevés impliquant l'arrivée des paquets audio après leur temps de restitution (*playout point*).

Les pertes dues aux liens de transmission sont rares surtout sur des réseaux dorsaux (*backbone networks*). Les applications utilisent le numéro de séquence du protocole de transport RTP afin de détecter la perte de paquets. Le protocole RTP (voir Annexe A) ne garantit pas l'arrivée des paquets à la destination, des mécanismes de contrôle de ces pertes sont alors nécessaires pour maintenir une qualité audio acceptable. Ils seront décrits au niveau du chapitre suivant.

3.4 Caractéristiques générales du délai et des pertes

3.4.1 Corrélation entre délai et pertes

En [KSB98], Kostas et Borella montrent que les cycles de délais et de pertes présentent des similarités sur des intervalles de temps longs (minute) et qu'ils sont caractérisés par une grande auto-corrélation, i.e., une dépendance avec leurs valeurs précédentes. Le premier résultat est intuitif car les pertes ont lieu généralement à cause d'une congestion, la congestion entraîne

des délais importants. Cette corrélation a certaines implications sur les applications temps-réel. La succession de délais longs est probablement suivie par des pertes, le modèle de perte en rafale cause une dégradation de la qualité d'une transmission audio sur Internet et affecte la performance des mécanismes de contrôle de la gigue et de la reconstruction des paquets perdus.

3.4.2 Dépendance temporelle des pertes

Le modèle de Gilbert

Un modèle de Markov à deux états : Modèle de Gilbert (Figure 3.2), est considéré en [SCK00] pour modéliser la dépendance temporelle des pertes. p est la probabilité que le paquet suivant soit perdu sachant que le précédent est arrivé à la destination. q est la probabilité que le paquet suivant arrive à la destination étant donné que le précédent était perdu. $p+q < 1$.

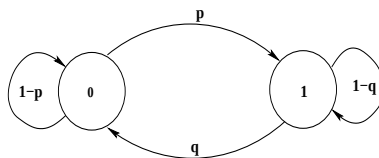


FIG. 3.2 – Modèle de Gilbert

Si $p+q=1$ le modèle de Gilbert est réduit à un modèle de Bernoulli. Ce dernier modèle suppose l'indépendance entre deux événements successifs, mais sur l'Internet la perte de paquet est une indication d'un début de congestion du réseau, avec une grande probabilité, le paquet suivant sera perdu [Jia99]. D'où la dépendance temporelle des pertes.

3.5 Conclusion

On a identifié dans ce chapitre les problèmes majeurs auxquels la transmission audio sur le réseau Internet est confronté, à savoir le délai, la gigue et la perte de paquets, puis on a présenté les caractéristiques générales des délais et des pertes qui doivent être tenues en compte par les mécanismes de contrôle d'une transmission audio sur l'Internet. En se basant sur ces points on étudiera au Chapitre 4 les différentes approches ayant été proposées pour permettre à une application d'audioconférence ou de téléphone sur Internet, de maintenir une qualité auditive acceptable même dans des cas où le réseau est assez congestionné.

Chapitre 4

Contrôle de la transmission audio sur l'Internet

4.1 Introduction

Les problèmes majeurs de la transmission de l'audio sur Internet ayant été analysés au chapitre précédent, on étudie dans ce chapitre les schémas les plus importants ayant été proposés afin de réduire l'effet des limitations causées par le réseau Internet. On présente précisément les mécanismes de contrôle de la gigue et de la perte des paquets.

4.2 Mécanismes de contrôle de la gigue

Le protocole RTP responsable du transport des données temps-réel ne garantit pas de contrôle de l'effet de la gigue. Ce dernier peut être alors réduit au niveau de l'application en maintenant les paquets reçus dans un buffer (*playout buffer*) pendant un certain délai avant de les jouer¹ (voir Figure 4.1). Ce délai doit être suffisamment petit pour ne pas affecter l'interactivité et suffisamment grand pour assurer la fluidité de la voix à la réception. De nombreux algorithmes décrits dans la littérature [Sch92][RGP98] choisissent différents modes d'opération le long de ce compromis. La valeur de ce délai peut être constante durant toute une communication audio ou être ajustée dynamiquement d'une période d'activité de la source (*talkspurt*) à une autre. Cependant, les fluctuations du délai de bout en bout (*end-to-end delay*) caractérisant l'Internet impliquent une qualité audio non satisfaisante pour un délai de compensation de la gigue constant.

Dans la suite on décrit un schéma de contrôle proposé en [Gar96] et qui est basé sur une estimation dynamique du délai de compensation en fonction du délai du réseau observé. On considère les notations suivantes (Figure 4.2) :

t_i l'instant auquel la source envoie le paquet i

a_i l'instant auquel le récepteur reçoit le paquet i

p_i l'instant auquel le paquet i est joué par le récepteur

n_i le délai (end-to-end transmission delay), i.e. $n_i = a_i - t_i$

b_i le délai de compensation de la gigue

¹c'est à dire envoyer le paquet sur un *driver* audio puis sur un haut parleur.

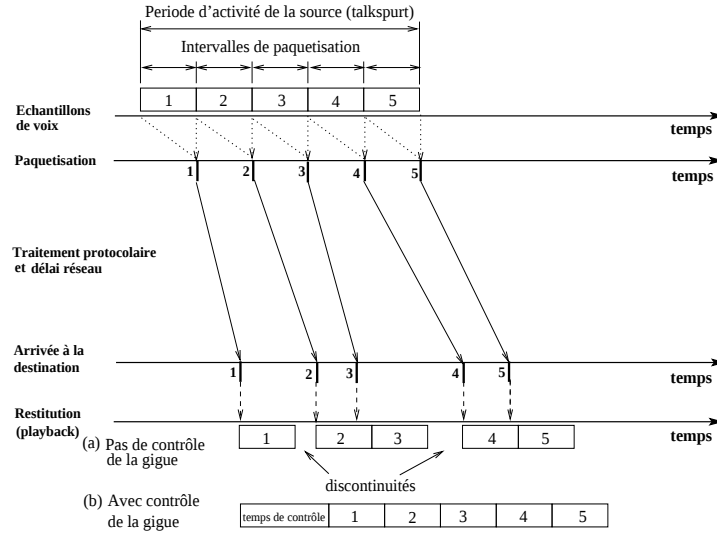


FIG. 4.1 – Schéma général du contrôle de la gigue

d_i le temps total dès que le paquet est généré par l'émetteur jusqu'à ce qu'il est joué par le récepteur, $d_i = n_i + b_i$

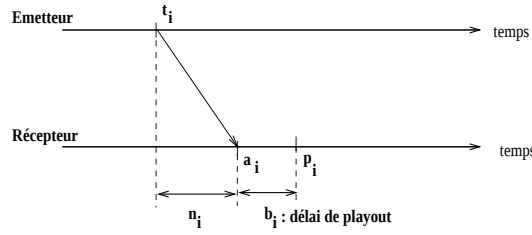


FIG. 4.2 – Délai de compensation de la gigue

Si i est le premier paquet d'une rafale², l'instant p_i est choisi selon la formule suivante :

$$p_i = t_i + \hat{d}_i + 4 * \hat{v}_i$$

où $\hat{d}_i$ représente la moyenne du délai et $\hat{v}_i$ sa variance estimées ($\hat{d}_i$ et $\hat{v}_i$ peuvent être estimées en utilisant un algorithme de gradient stochastique).

Si i n'est pas le premier paquet, l'instant p_i est déterminé selon la formule :

$$p_i = p_j + t_i - t_j$$

où j dénote le premier paquet de la rafale à laquelle i appartient.

L'algorithme de base du calcul de b_i suppose que les délais des paquets dans le réseau sont Gaussiens. Il choisit b_i de telle sorte que d_i soit égal au délai moyen $\hat{d}_i$ additionné de 4 fois la variance du délai $\hat{v}_i$. Un résultat de la distribution gaussienne dit que la probabilité que le délai d'un paquet quelconque dépasse $\hat{d}_i + 4 * \hat{v}_i$ est très faible [Gar96].

²Rafale ou période d'activité (talkspurt) : segment de parole qui se trouve entre deux silences.

Ce schéma considère deux hypothèses :

1. Le maintien d'une synchronisation entre les horloges système de l'émetteur et du récepteur assurée généralement par le protocole NTP (*Network Time Protocol*) [RFC1305].
2. Les délais des paquets audio sur le réseau suivent une distribution gaussienne.

Ces deux hypothèses ne sont pas toujours évidentes. La synchronisation basée sur NTP peut ne pas être assez précise pour interopérer avec la nature temps-réel du processus de génération/restitution audio : des valeurs de précision de quelques centaines de millisecondes peuvent paraître si une machine cliente n'est pas connectée directement à un serveur NTP primaire mais à un serveur secondaire et à travers un lien congestionné. En plus l'hypothèse de la distribution gaussienne des délais n'est pas vraie dans le cas de congestion du réseau, des piques de délais peuvent avoir lieu dans ce cas [BVG96].

Rocchetti et al. proposent en [RGP98] un mécanisme de contrôle du délai de compensation de la gigue qui permet de résoudre les problèmes que peut confronté le schéma décrit en [Gar96]. Ce schéma assure une synchronisation précise des horloges émetteur/récepteur basée sur une estimation du RTT³ et permet d'adapter l'estimation du temps de restitution des paquets audio sur chaque rafale (*talkspurt*) avec un minimum de surcoût. L'idée est simple : quand l'émetteur transmet le premier paquet d'une rafale audio il met dans ce paquet une estampille de temps (*timestamp*) qu'on notera H et qui correspond à la valeur de son horloge lue à cet instant. A la réception de ce premier paquet, la destination corrige son horloge qu'on note H_R avec la valeur de H reçue.

$$H_R = H$$

Les paquets suivants appartenant à la même rafale contiennent eux aussi la valeur de l'horloge de l'émetteur lue à l'instant de leur envoi. Quand ces paquets arrivent à la destination, leur estampille est comparée à la valeur de l'horloge du récepteur, si la valeur de l'estampille est égale à celle de l'horloge, le paquet est immédiatement joué (*played out*). Si le *timestamp* est supérieur à la valeur de l'horloge du récepteur ce paquet est bufférisé et sera joué après un temps égal à la différence entre la valeur de l'estampille et celle de l'horloge. Finalement si la valeur de l'estampille est attachée au paquet est inférieure à la valeur de l'horloge du récepteur, le paquet est éliminé et ne peut pas être présenté vu le retard de son arrivée.

4.3 Mécanismes de résolution des pertes

Le contrôle d'erreur d'une transmission de données ayant des propriétés temps-réel ne peut pas être réalisé par des mécanismes de bout en bout comme ceux utilisés par les protocoles traditionnels pour effectuer un transfert de fichiers par exemple. Le service du réseau Internet ne peut pas garantir simultanément une transmission fiable et des délais bornés, les retransmissions causent des délais arbitraires sur la transmission. Deux types d'approches ont été proposés afin de minimiser le taux de perte et ne pas introduire de latence sur une transmission audio, ils sont discutés dans les deux paragraphes qui suivent.

³Round Trip Time : délai aller-retour

4.3.1 Schémas de la FEC (*Forward Error Correction*)

Plusieurs résultats [ABR01][BPT99] ont montré l'utilité des schémas de contrôle d'erreur basés sur les mécanismes de la FEC pour diminuer l'effet de la perte de paquets sur la qualité audio. Ce mécanisme consiste à envoyer une information de redondance conjointement au flux audio original, ce qui augmente la probabilité de résoudre la perte de certains paquets. Par exemple les paquets audio originaux sont codés en PCM et l'information de redondance en LPC ou en GSM. La Figure 4.3 décrit un schéma simple de FEC consistant à l'envoi dans chaque paquet n d'une information de redondance concernant le paquet $n-1$, d'autres schémas proposent l'envoi d'une information des paquets $n-1$ et $n-2$ dans le but de remédier à la perte de deux paquets successifs.

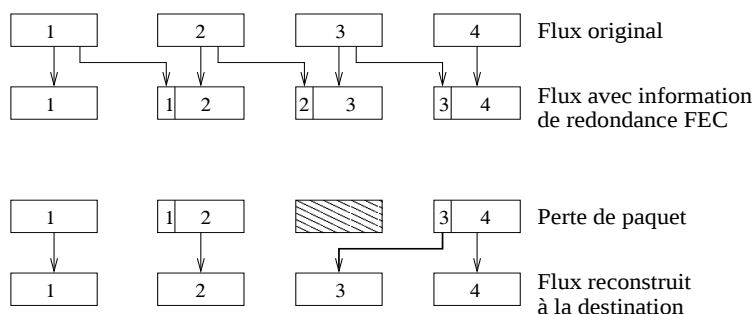


FIG. 4.3 – Schéma simple du mécanisme de FEC :

Un paquet n contient une information de redondance du paquet $n-1$

Mais le problème c'est que l'information de FEC augmente aussi la bande passante utilisée et par conséquent augmente la probabilité de perte. D'où la nécessité de coupler le schéma de FEC avec un schéma de contrôle s'appuyant sur une information reflétant l'état du réseau comme le montre la Figure 4.4.

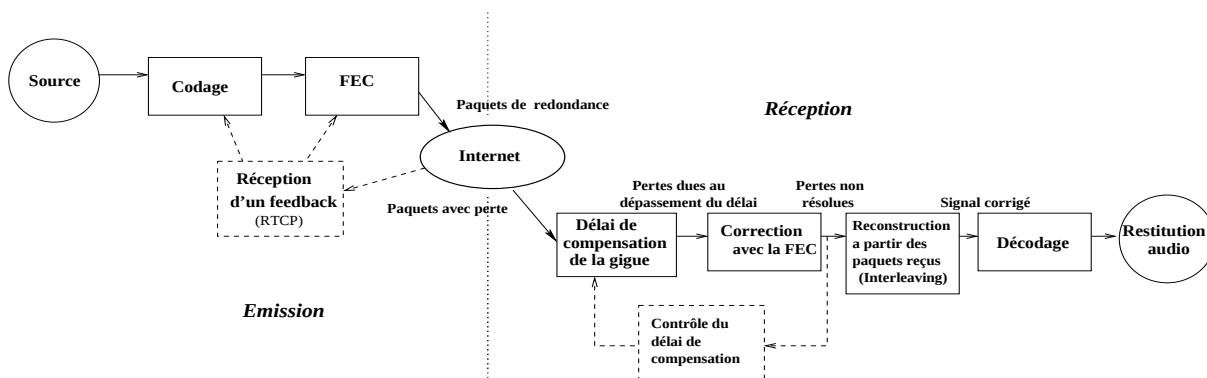


FIG. 4.4 – Schéma de transmission audio avec contrôle d'erreur adaptatif

Bolot et al. montrent en [BPT99] que trouver la meilleure information de FEC sous certaines contraintes réseau est un problème d'optimisation, et proposent un algorithme qui combine les

schémas de contrôle de débit et de redondance dans le but d'optimiser la qualité audio perçue à la destination en s'adaptant aux conditions réseau estimées à partir des *feedback* de RTCP (c.f. Annexe A). Cet algorithme adaptatif assure une grande performance, mais ne permet pas de garantir une bonne qualité dans certaines conditions réseau vu qu'il utilise un modèle de Bernoulli pour modéliser les pertes, ce modèle n'est pas bien adéquat pour le service "*best effort*" de l'Internet caractérisé par des pertes variables et des pertes pouvant avoir lieu en rafales (*bursty losses*).

4.3.2 Mécanismes de contrôle à boucle fermée

Dans les schémas de contrôle à boucle fermée ARQ (*Automatic Repeat Request*) le récepteur demande à l'émetteur de retransmettre les paquets perdus détectés par les numéros de séquence manquants. Ces mécanismes ne peuvent pas être appliqués pour la transmission de la voix sur l'Internet vu les besoins temps-réel que nécessite l'interactivité de l'application. Elles sont plus intéressantes dans le cas d'applications comme l'audio/vidéo à la demande n'ayant pas de contraintes temps-réel strictes. Cependant l'émetteur peut combiner les schémas de FEC et de ARQ : la FEC est utilisée quand de grandes pertes apparaissent, l'ARQ est appliqué pour réduire le taux de perte et limiter la propagation d'erreur⁴.

En [DLW93] une nouvelle approche pour la résolution des pertes de paquets audio sur des réseaux haut débit S-ARQ (*Slack Automatic Repeat Request*) est introduite. Comme le montre la Figure 4.5 ce schéma est basé sur la retransmission de paquets perdus et permet d'étendre le temps de contrôle de la gigue d'une rafale de paquets en attente de recevoir un paquet retransmis.

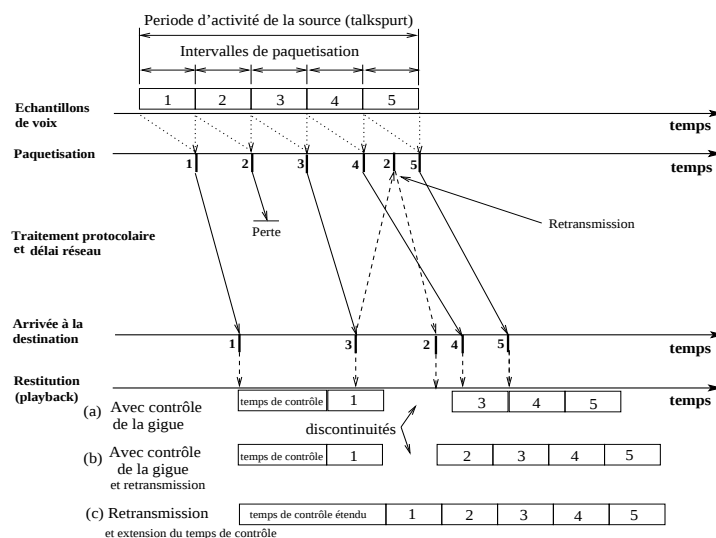


FIG. 4.5 – Schéma de transmission d'une rafale audio en présence de perte

Dans la Figure 4.5, le paquet 2 est considéré comme perdu, le scénario (a) montre le comportement des protocoles sans retransmission, le paquet perdu n'est pas joué à la destination ce qui cause une discontinuité de durée égale au temps de paquetisation. Dans les scénarios

⁴Pour les codecs G.723.1 et G.729, la perte d'un paquet influe sur les paquets suivants.

(b) et (c), le récepteur demande la retransmission du paquet 2 détecté perdu à la réception du paquet 3. Le scénario (b) montre l'inconvénient de l'utilisation d'un schéma de contrôle ordinaire qui ne corrige pas la discontinuité introduite. Le scénario (c) montre l'amélioration introduite par l'extension du temps de contrôle basée sur une estimation du RTT du réseau.

Un point négatif de ce schéma c'est qu'il augmente le délai de bout en bout de tous les paquets audio, ce qui nécessite la considération d'un seuil de suppression des retransmissions.

4.4 Conclusion

On a présenté au niveau de ce chapitre deux mécanismes de contrôle de la gigue, le premier schéma basé sur une estimation dynamique du délai de compensation de la gigue en fonction du délai réseau mesuré nécessite le maintien d'une synchronisation entre les horloges émetteur/récepteur qui utilise le protocole NTP, chose qui n'est pas toujours évidente. Le deuxième schéma est plus simple de point de vue traitement et mise en oeuvre et ne suppose pas la synchronisation entre l'émetteur et le récepteur. Deux schémas de contrôle d'erreur ont été analysés : le schéma préventif de la FEC permet de remédier à la perte de certains paquets mais souffre du problème de pertes en rafales et du gaspillage de la bande passante. Les schémas à boucle fermée bien que limités par les contraintes d'interactivité des applications temps-réel peuvent offrir de bonnes performances dans le cas de réseaux haut débit caractérisés par des délais faibles.

Dans le Chapitre 5, on présentera les résultats d'expérimentations de transmission de flux audio qu'on a effectué sur un réseau expérimental à haut débit.

Chapitre 5

Expérimentations

5.1 Introduction

Dans ce chapitre on présente les résultats d'un ensemble d'expérimentations effectuées sur un réseau à très haut débit (VTHD : Vraiment Très Haut Débit) caractérisé par une bande passante abondante grâce à l'utilisation de supports de transmission optiques et aux technologies de multiplexage en longueur d'onde WDM.

Un outil de génération de trafic audio a été développé afin d'effectuer des mesures réelles et d'étudier le comportement de la transmission de plusieurs flux audio dans un réseau IP sans réservation de ressources.

Les expérimentations ont été effectuées entre deux machines connectées au réseau VTHD, une située à l'INRIA Sophia Antipolis et l'autre à l'INRIA Rocquencourt (Paris).

5.2 Environnement des tests

5.2.1 Le réseau VTHD

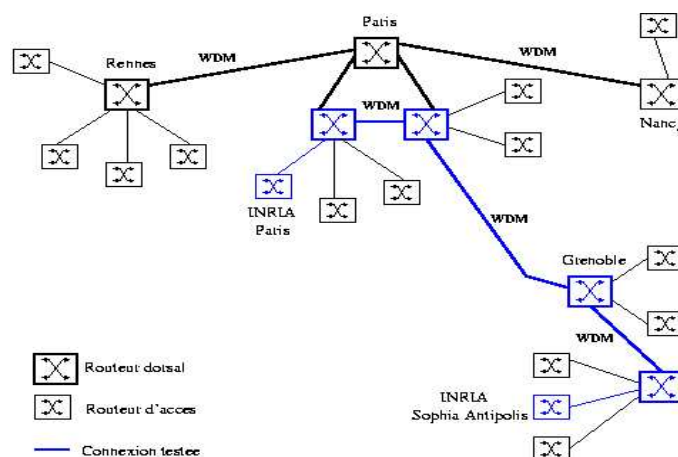


FIG. 5.1 – Réseau VTHD testé

Le réseau dorsal VTHD (Figure 5.1) comporte huit routeurs interconnectés par des canaux

optiques à 2,5 Gbit/s. Sur ce réseau dorsal sont raccordés 18 routeurs d'accès. Le réseau VTHD est pour le présent un réseau fermé : il offre uniquement la connectivité entre les sites VTHD.

5.2.2 La connexion de bout en bout testée

La connexion que j'ai utilisé pour effectuer des expérimentations de bout en bout comporte, entre autre, deux machines de type PC sur lesquelles j'ai installé l'ensemble des outils nécessaires pour les tests. Afin de pouvoir mesurer le délai de transmission de bout en bout, les horloges système des deux machines ont été synchronisées avec des serveurs NTP (Figure 5.2)

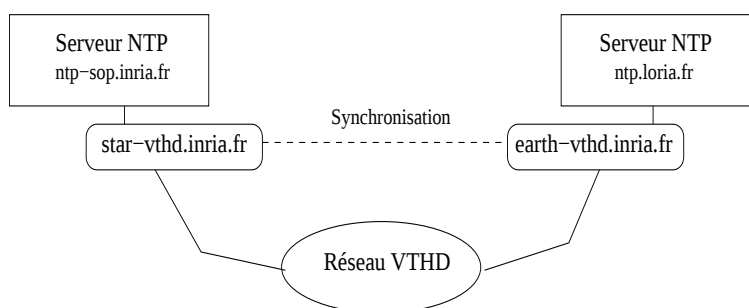


FIG. 5.2 – Connexion testée : synchronisation des horloges des deux machines

5.2.3 Outils utilisés pour les tests

L'outil audio RAT (*Robust Audio Tool*)

RAT [RAT] est un outil qui transmet et reçoit le canal son qui peut être utilisé entre une source et une destination ou entre plusieurs participants d'une audio conférence (dans ce travail on s'est intéressé à une transmission en point à point). Il permet d'utiliser les différents types de codage (PCM, GSM, LPC...) décrits en Section 2.2 présentant des caractéristiques (débit, qualité audio) différentes ainsi que des mécanismes de reconstruction des paquets perdus. RAT se base sur les standards IETF [RFC1890] et utilise le protocole RTP en-dessus de UDP/IP. La version qui a été utilisée dans ces expérimentations est *rat-4.2.20*.

Les outils RTP TOOLS

C'est un ensemble d'outils (rtpdump, rtpplay, rtpsend, rtptrans) [Tools] permettant l'écoute des paquets RTP et RTCP sur un port donné d'adresse déterminée et génère un fichier qui peut être rediffusé sur le réseau.

Le générateur de flux audio développé

Afin d'étudier le comportement d'une transmission de plusieurs flux audio pouvant circuler sur le réseau, j'ai développé pendant ce stage un outil (utilisant le langage Perl) permettant l'envoi d'un nombre de flux de sources audio, donné comme paramètre, vers une adresse et un

port de destination. Il utilise l'outil *rtpplay* pour envoyer un fichier d'écoute d'une transmission audio enregistré par l'outil *rtpdump*.

5.3 Expérimentations et analyse des résultats

5.3.1 Mesure du débit des différents codecs

Le but de cette première expérience est de mesurer le débit réel engendré par un seul flux audio pour les trois types de codage différents PCM, LPC et GSM présentant des débits théoriques respectifs de 64 kbps, 5,6 kbps et 13 kbps. Ce test a été effectué en utilisant l'outil RAT lancé sur les deux machines, un enregistrement audio est utilisé comme fichier d'entrée pour RAT. Trois transmissions ont été effectuées en modifiant à chaque fois le codec à utiliser par l'outil RAT. Le Tableau 5.1 montre les caractéristiques de ces codecs.

Codage	Fréquence d'échantillonnage	Débit	Codec
PCM	8000 Hz	64 kbps	ITU G.711
LPC	8000 Hz	5,6 kbps	Pitch excited linear prediction codec
GSM	8000 Hz	13,2 kbps	Full rate GSM speech codec

TAB. 5.1 – Codecs utilisés pour les tests

La Figure 5.3 montre le débit mesuré sur la machine réceptrice du flux audio. Cette figure montre que le flux audio PCM engendre un débit réel supérieur à 100 kbps. Les codages GSM et LPC engendrent des débits moins élevés que le PCM (inférieurs à 100 kbps) mais qui sont importants par rapport à leur débits théoriques.

Ce résultat peut être expliqué par le trafic introduit par les entêtes protocolaires RTP/UDP/IP de 12 octets, 8 octets et 20 octets respectivement. Pour le codage PCM la taille des paquets audio utilisée est de 160 octets (un paquet est généré toutes les 20 ms à un débit de 64kbps). Avec les 40 octets d'entêtes ajoutées la taille d'un paquet audio transmis sur l'Internet sera de 200 octets, ce qui forme un débit réel de 80 kbps. Pour le codage GSM, la taille des paquets audio est de 33 octets (un paquet est généré chaque 20 ms à un débit de 13,2 kbps). La taille des entêtes ajoutées est plus grande que le contenu audio du paquet ce qui résulte en un débit réel de 29,2 kbps. L'inefficacité des codages à bas débits est encore plus importante dans le cas du codage LPC à 5,6 kbps, la taille des paquets étant de 14 octets, le débit réel est 21,6 kbps, ce débit représente le débit minimum d'une transmission de paquets audio sur l'Internet. On peut alors résumer les débits réels des différents codages dans le Tableau 5.2.

Codage	Débit	Débit réel (avec entêtes RTP/UDP/IP)
PCM	64 kbps	80 kbps
LPC	5,6 kbps	21,6 kbps
GSM	13,2 kbps	29,2 kbps

TAB. 5.2 – Débit réel des codecs audio

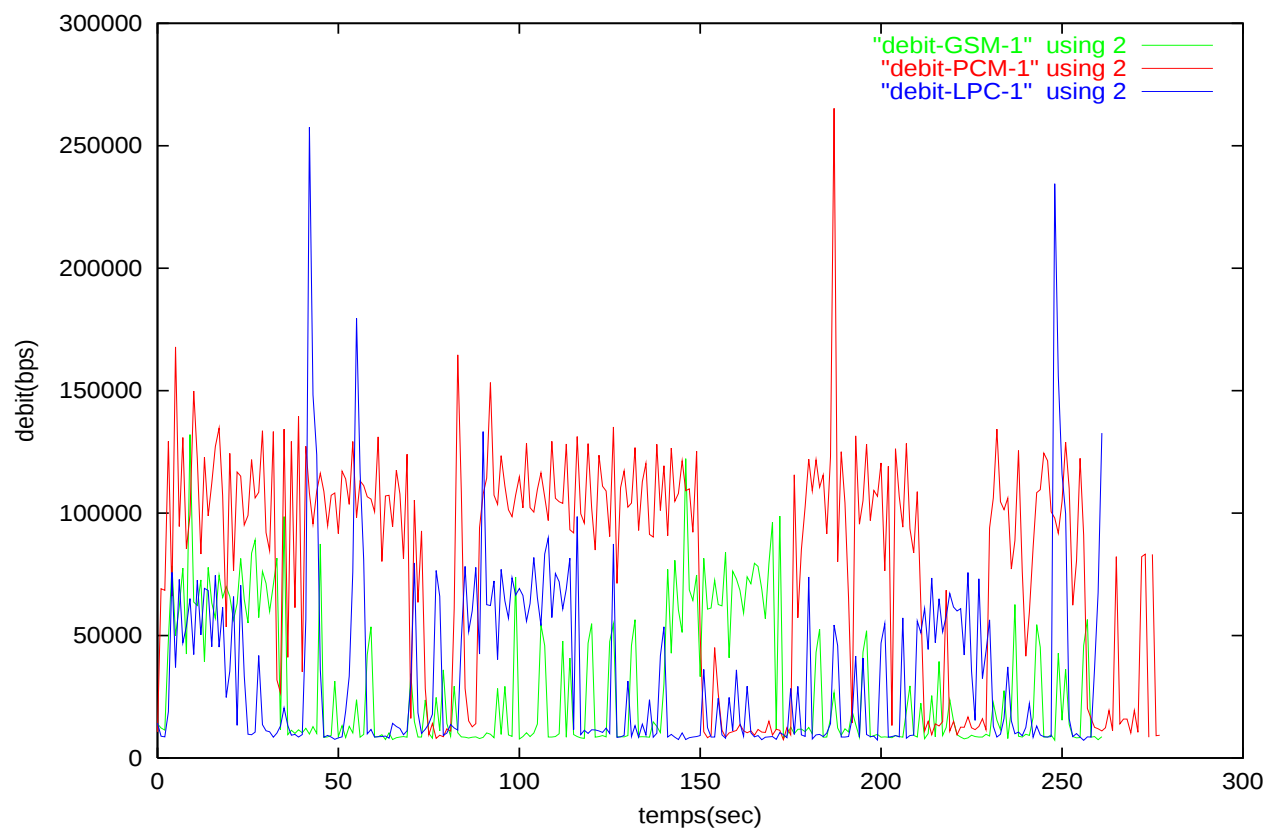


FIG. 5.3 – Débit d'une transmission audio avec différents codecs

5.3.2 Débit d'une transmission de plusieurs flux audio

Afin de mesurer les caractéristiques d'une transmission d'un grand nombre de flux audio sur le réseau on a utilisé le générateur de trafic audio qu'on a développé avec des fichiers audio enregistrés par l'outil *rtpdump* correspondants à des transmissions audio avec les codages PCM, LPC, et GSM. La figure 5.4 montre le débit pour la transmission de deux sources audio avec les trois types de codage. Les Figures 5.5 et 5.6 montrent la même chose pour le cas d'une transmission de 5 et 10 sources respectivement. L'outil développé pour la génération de trafic étant basé sur l'appel système *fork()* permettant de créer des processus qui envoient le trafic audio, est limité par les ressources système des machines utilisées.

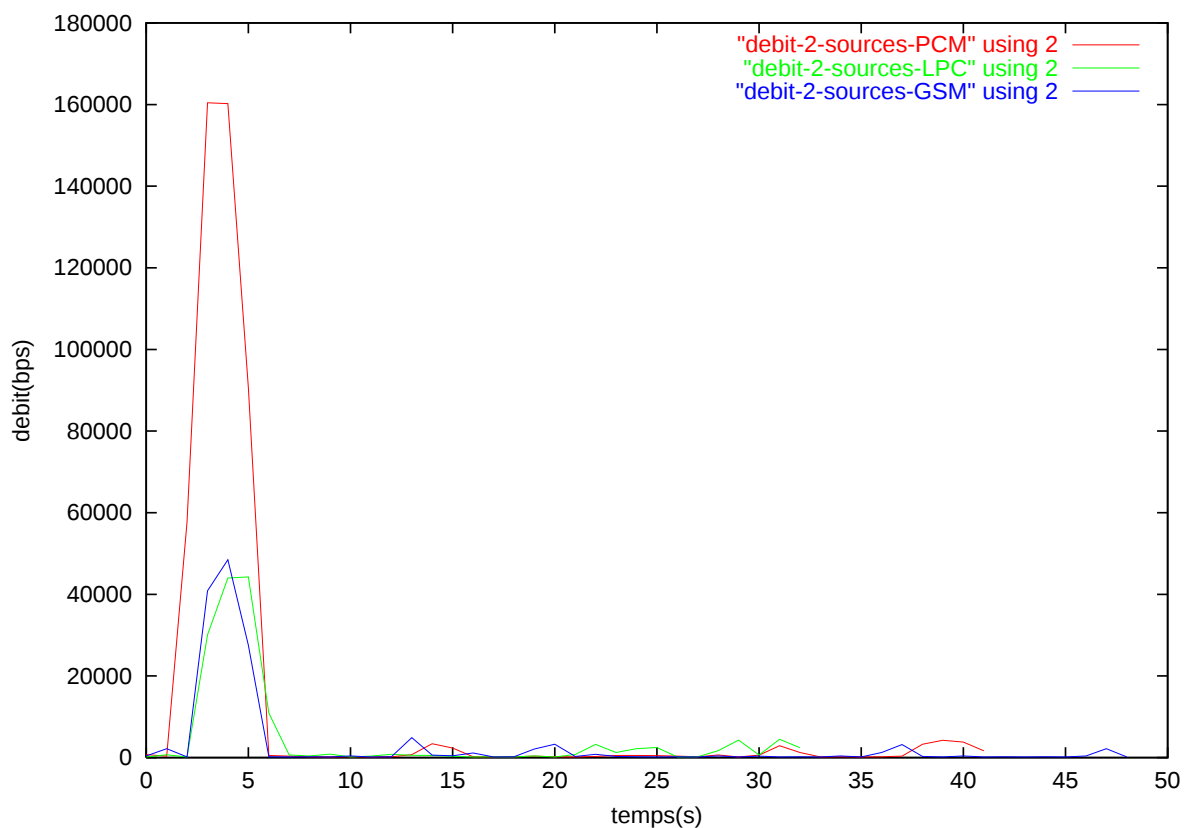


FIG. 5.4 – Débit pour 2 sources audio

Dans le Tableau 5.3 on résume les débits maximums engendrés lors de l'envoi de plusieurs trafics audio avec les différents codages.

	2 sources	5 sources	10 sources
PCM	160 kbps	400 kbps	880 kbps
GSM	48 kbps	120 kbps	250 kbps
LPC	42 kbps	110 kbps	240 kbps

TAB. 5.3 – Débit pour une transmission de plusieurs flux audio

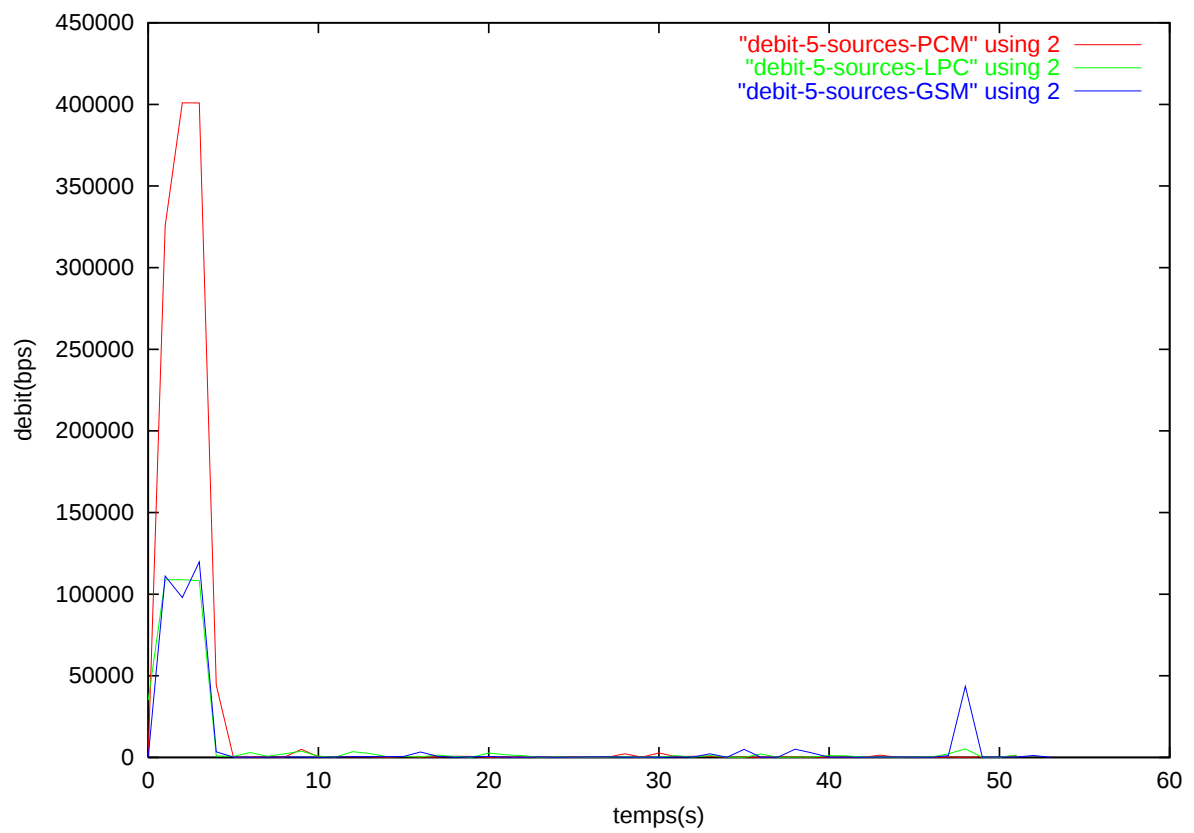


FIG. 5.5 – Débit pour 5 sources audio

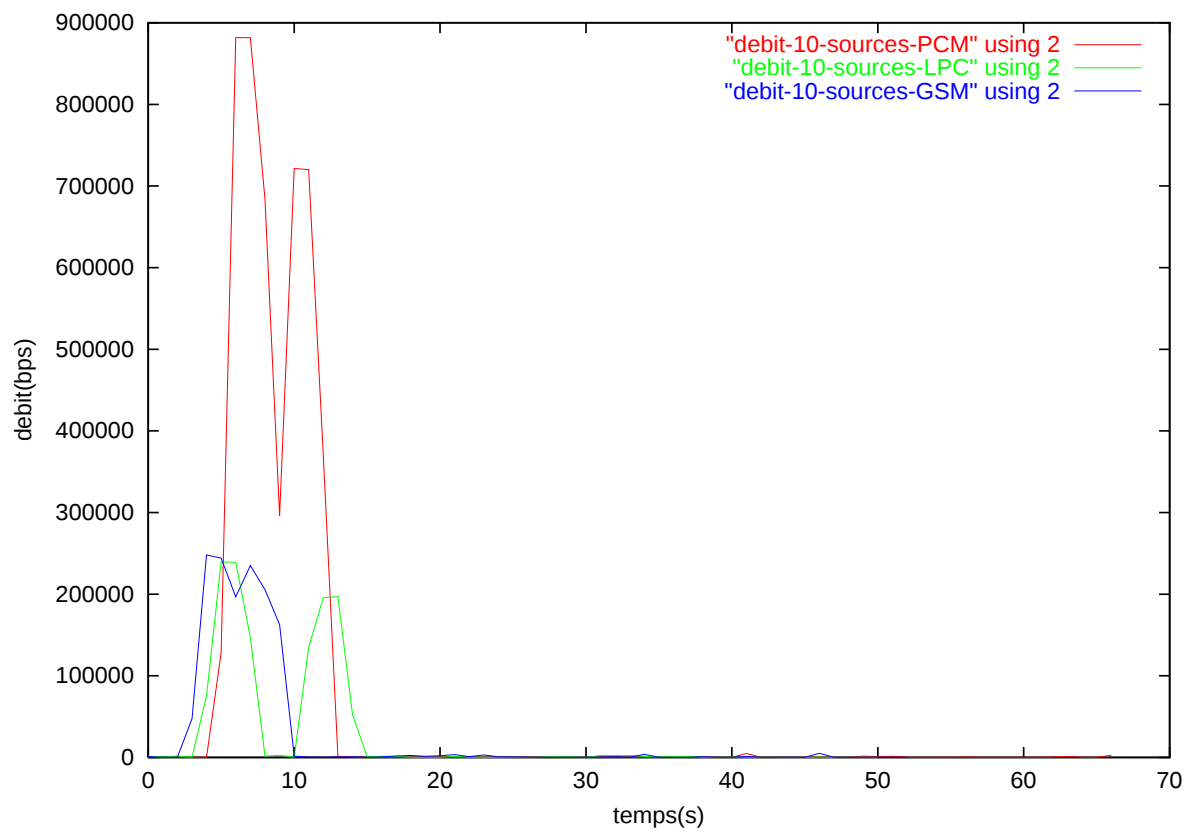


FIG. 5.6 – Débit pour 10 sources audio

Les valeurs de débits mesurées sont en accord avec les valeurs de débit réel pour un seul flux décrites au Tableau 5.2. Dans le cas de plusieurs sources audio, le débit engendré est égal au débit réel d'un seul flux multiplié par le nombre de sources. Les débits engendrés par le codage PCM restent les plus importants, ceux du LPC sont les moins élevés. Dans le cas d'un grand nombre de sources audio, des centaines par exemple, le débit total d'un codage PCM sera de quelques dizaines de Mbps, ce qui présente une charge importante de bande passante du réseau. D'où surgit l'intérêt de l'utilisation des mécanismes de multiplexage RTP permettant de réduire le surcoût des entêtes en ajoutant une seule entête IP/UDP/RTP pour plusieurs paquets audio, cependant ces mécanismes augmentent les délais de transmission des paquets.

5.3.3 Délais de transmission audio sur le réseau haut débit

Dans cette section on s'intéresse à l'estimation du délai de bout en bout de la transmission audio sur le réseau haut débit testé. Pour ceci on a utilisé l'outil *tcpdump* lancé sur la machine d'émission pour avoir le temps d'envoi sur le réseau des paquets audio (t_i), l'outil *rtpdump* lancé sur la machine réceptrice fournit le temps d'arrivée des paquets (r_i). Les horloges des deux machines ayant été synchronisées avec des serveurs NTP, le délai réseau relatif à un paquet i est calculé par la formule :

$$d_i = r_i - t_i$$

La Figure 5.7 montre le délai réseau d'un flux (la transmission effectuée est de 2 flux), qu'on a calculé à partir des fichiers de traces. D'après la courbe de délai tracée les délais réseau des trois codages (PCM, LPC et GSM) sont très proches, elles ont une valeur de 4ms. Le délai engendré par le codage PCM est un peu plus élevé que 4ms. La différence entre les délais engendrés s'explique par les taux de compression qui diffèrent pour les trois codecs : la taille des paquets PCM est de 160 octets, celle des paquets GSM est de 33 octets et celle des paquets LPC est de 14 octets, le délai de transmission est proportionnel à la taille des paquets.

La faible valeur du délai réseau (4ms) s'explique par le délai de propagation très réduit grâce aux technologies de transmission optique ($5\mu s/Km$) [Jia99].

Le réseau n'étant pas chargé lors des tests effectués, le délai mesuré ne présente pas de grandes fluctuations. Pour ces expérimentations on n'a pas détecté des pertes de paquets, ceci est expliqué par la bande passante énorme des liens réseau et la grande capacité des routeurs traversés par la connexion.

Les délais de l'émetteur et du récepteur étant d'environ 20 ms pour les trois types de codage, le délai de bout en bout total constitué de la somme du délai de l'émetteur, du délai réseau et du délai du récepteur sera d'une valeur de 44ms. Cette valeur permet d'avoir une très bonne qualité audio comme s'est expliqué au Tableau 5.4 [DLW93].

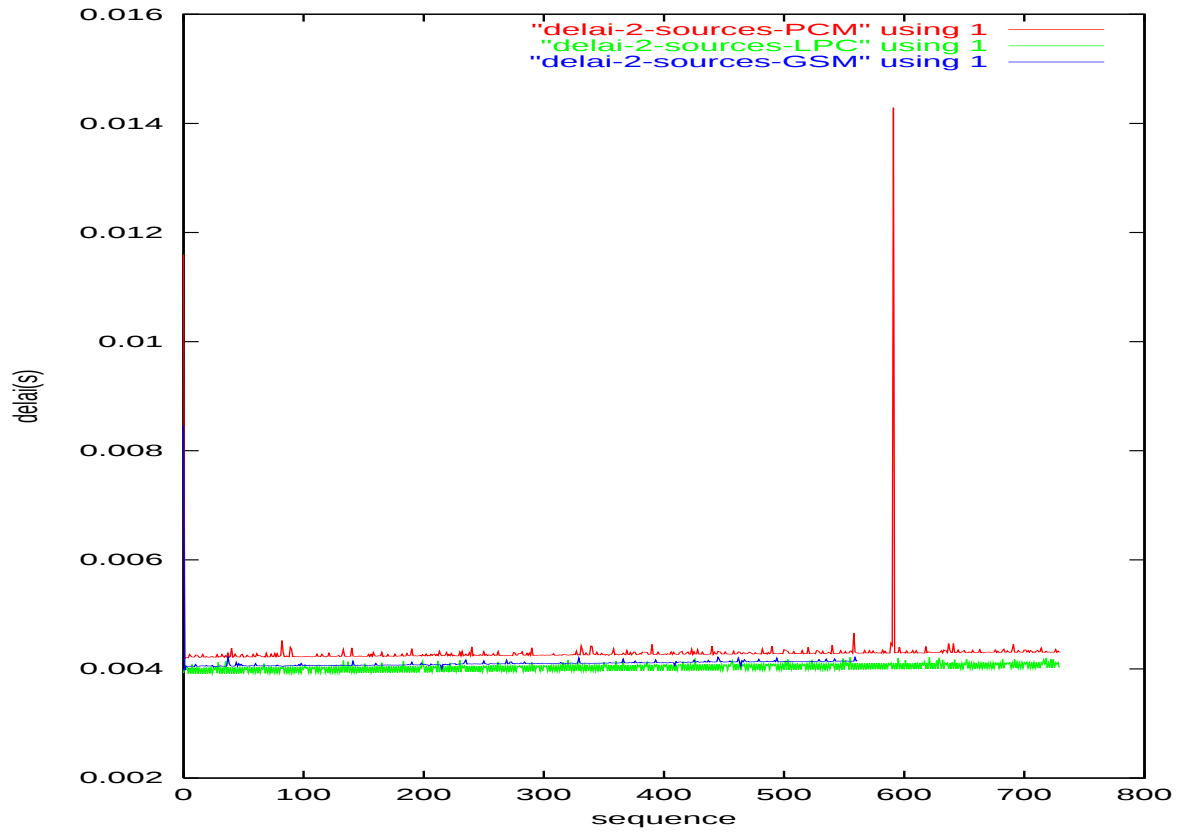


FIG. 5.7 – Délai réseau d'un flux d'une transmission de 2 sources audio

Délai de bout en bout	Effet sur la qualité de la voix
> 600 ms	La voix devient incohérente et non intelligible
600 ms	La voix est à peine cohérente
250 ms	L'interactivité d'une conversation est affectée par les longs délais
100 ms	La voix est cohérente et intelligible
50 ms	Bonne qualité

TAB. 5.4 – Effet du délai de bout en bout sur la qualité de l'audio

5.4 Conclusion

Dans ce chapitre on a présenté un ensemble d'expérimentations de transmission audio effectuées sur un réseau de test à haut débit. D'après les mesures de débit et de délais effectuées pour la transmission d'un ou de plusieurs flux audio on peut conclure que le trafic engendré les entêtes IP/UDP/RTP présente un surcoût de la transmission audio sur l'Internet et un gaspillage de la bande passante réseau. Le délai réseau très réduit sur le réseau haut débit testé offre l'avantage d'avoir un délai de bout en bout en dessous de 200ms qu'impose l'interactivité des applications audio.

Chapitre 6

Conclusion et perspectives

Durant ce stage on a étudié les différents aspects que fait intervenir une application audio sur l'Internet, à savoir le codage, la paquetisation et la transmission des données audio. Ensuite, on a identifié les problématiques de la transmission audio sur le réseau Internet "best effort" qui sont principalement le délai de bout en bout, la gigue et la perte des paquets. En se basant sur ces points, on a analysé et comparé les différentes approches de contrôle de la gigue et des pertes proposées pour permettre à une application d'audioconférence ou de téléphonie sur Internet, de maintenir une qualité auditive acceptable même dans le cas où le réseau est assez congestionné.

La majorité des études faites ont été basées sur la simulation ou sur des modèles théoriques simples. On a opté dans ce travail pour la réalisation de tests réels et de mesures sur un réseau expérimental à très haut débit. Ainsi, on a développé et installé une plate-forme de tests basée sur un générateur de trafic audio.

Nous pouvons conclure que :

- Les codecs audio, conçus à l'origine pour des réseaux à commutation de circuits, génèrent des paquets de taille constante à une cadence régulière ce qui n'est pas approprié aux réseaux à commutation de paquets comme l'Internet caractérisés par des variations imprévisibles des paramètres réseau.
- Le surcoût protocolaire des entêtes IP/UDP/RTP cause un gaspillage de la bande passante réseau d'où l'intérêt de multiplexer les flux audio sur un même paquet.
- Le délai réseau très faible caractérisant les réseaux haut débit permet d'avoir un délai de bout en bout en dessous des 200ms imposés par les besoins d'interactivité. Ces délais réduits permettront l'utilisation des mécanismes de contrôle d'erreur à boucle fermée qui sont plus efficaces vu les taux de perte faibles sur ces réseaux.
- La surabondance de bande passante rend d'autre part la compression des flux moins indispensable, l'absence de compression réduit le délai initial dû au codec, et permet de garantir plus facilement un délai total de l'ordre de 150 ms comparable à celui de des réseaux à commutation de circuits.

Dans nos travaux futurs on s'intéressera à :

- Trouver un modèle analytique permettant d'approcher le comportement des multiplexes de flux multimédias.
- Construire un générateur de trafic permettant d'intégrer les aspects de signalisation et d'effectuer des mesures réelles sur le réseau.

Annexe A

Format des paquets RTP/RTCP

Format des paquets RTP

L'entête d'un paquet RTP (Figure A.1) est obligatoirement constituée de 12 octets, éventuellement suivie d'une liste d'identificateurs de sources contributrices CSRCs. Cette entête précède le *“payload”* qui représente les données utiles.

La Figure A.1 montre l'entête des paquets RTP, dont une description sommaire sera faite dans la suite.

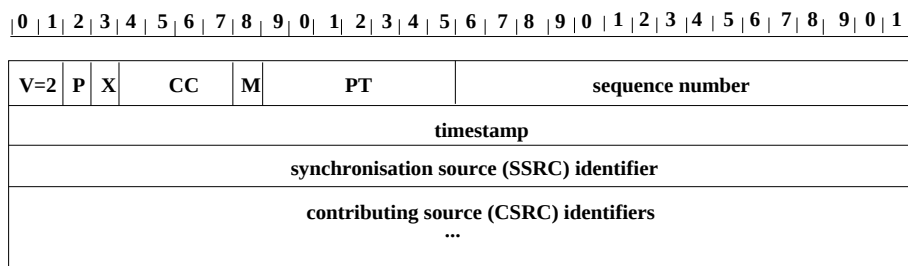


FIG. A.1 – Format de l'entête RTP

L'en-tête RTP comportera les informations suivantes :

- Le champ version **V** (2 bits) indique la version du protocole (la version actuelle est la version2).
- Le champ padding **P** (1 bit), si P est égal à 1, le paquet contient des octets additionnels de bourrage (*padding*) pour finir le dernier paquet.
- Le champ extension **X** (1 bit), si X=1 l'en-tête est suivie d'un paquet d'extension.
- Le champ CSRC count **CC** (4 bits) contient le nombre de CSRC qui suivent l'entête.
- Le champ marker **M** (1 bit), son interprétation est définie par un profil d'application (*profile*).
- Le champ payload type **PT** (7 bits) identifie le type du *payload* (audio, vidéo, image, texte, html, etc.)
- Le champ **sequence number** (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** (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** (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. 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** (32 bits), identifie les sources contributrices.

Format des paquets RTCP

Le protocole RTCP est basé sur des transmissions périodiques de paquets de contrôle par tous les participants dans la session. Il existe 5 types de paquets RTCP pour transporter des informations de contrôle :

SR (*Sender Report*), transmission de statistiques des participants actifs en émission.

RR (*Receiver Report*), transmission de statistiques des participants passifs.

SDS (*Source Description items*), paquet de description de la source (CNAME, NAME, EMAIL, PHONE,...)

BYE Fin de participation.

APP Fonctions spécifiques à l'application.

Chaque paquet RTCP commence par une partie fixe, suivie par des éléments structurés qui peuvent être de longueur variable selon le type de paquet, mais toujours terminé sur une frontière de 32 bits. La Figure A.2 décrit le format des paquets RTCP SR. La Figure A.3 montre les champs contenus dans un paquet RTCP RR.

L'en-tête RTCP comportera les informations suivantes :

- Le champ version **V** (2 bits).
- Le champ padding **P** (1 bits) indique qu'il y a du bourrage dont la taille est indiquée dans le dernier octet.
- Le champ reception report count **RC** (5 bits), nombre de compte-rendus dans le paquet.
- Le champ packet type **PT** (8 bits) 200 pour SR et 201 pour RR.
- 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).

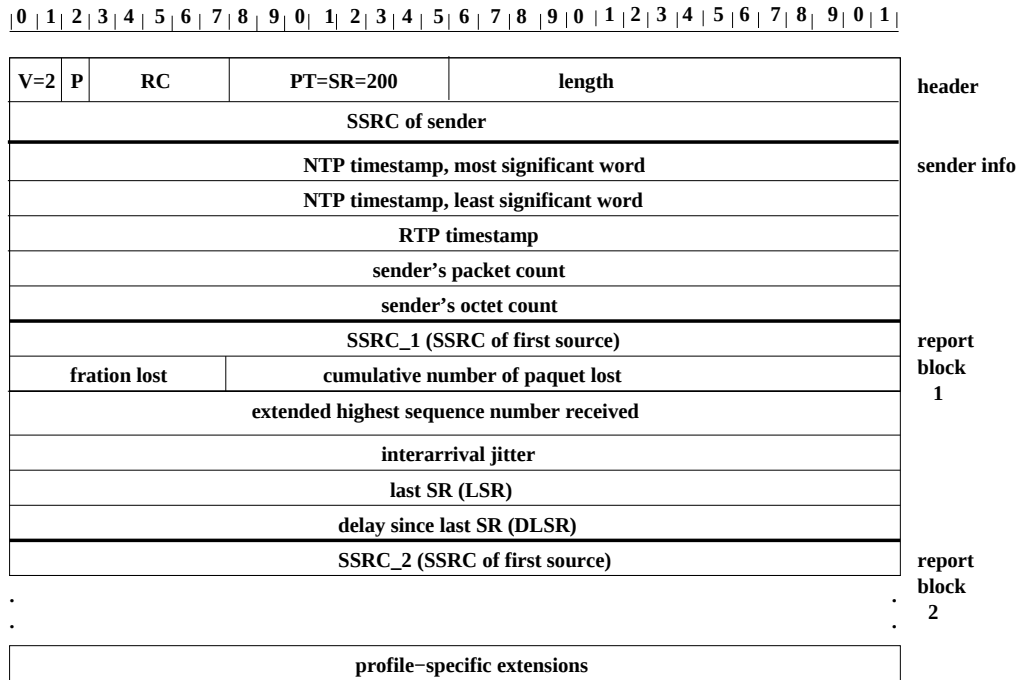


FIG. A.2 – Format d'un paquet RTCP SR

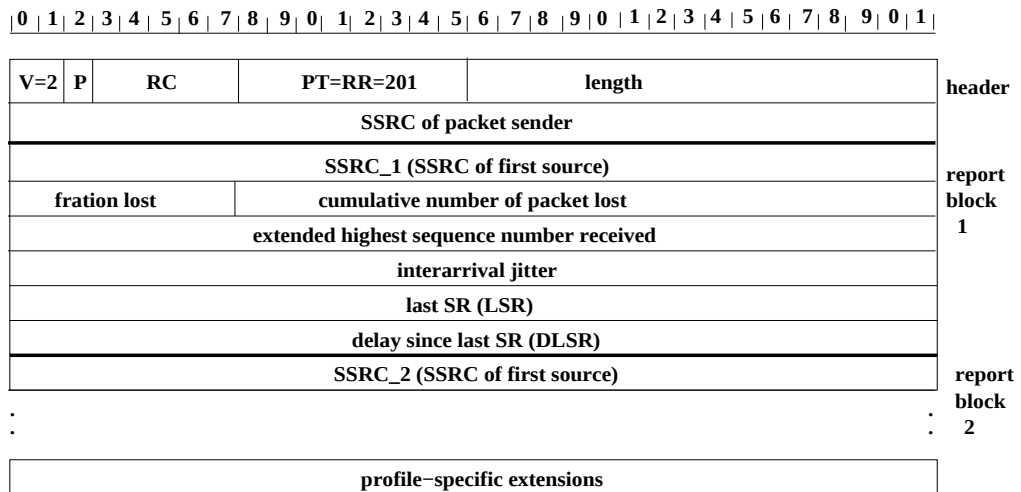


FIG. A.3 – Format d'un paquet RTCP RR

- 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. L'interarrival jitter est calculée à chaque packet de donnée reçu par la source `SSRC_n`. C'est en fait le temps relatif de transit entre deux paquets de données : $i - 1$ et i . La formule pour le calculer est [RFC1889] :

$$J = J + (| D(i - 1, i) | - J)/16$$

- Le champ ***last SR timestamp*** (32 bits).
- Le champ ***delay since last SR*** (32 bits).

Bibliographie

- [ABR01] E. Altman, C. Barakat, V. Ramos, “*On the utility of FEC mechanisms for audio applications*”, INRIA Sophia Antipolis, Institut Eurécom, Sophia Antipolis, France, 2001.
- [BPT99] J.C. Bolot, S.F. Parisi, D. Towsely, “*Adaptive FEC-Based Error Control for Internet Telephony*”, Proc. IEEE Infocom’99, New York, Mar. 1999.
- [Bra65] P. T. Brady, “*A Technique for Investigating On-Off Patterns of Speech*”, Bell System Technical Journal, Jan. 1965.
- [Bra71] P. T. Brady, “*Effects of Transmission Delay on Conversational Behavior on Echo-Free Telephone Circuits*”, Bell System Technical Journal, Jan. 1971.
- [BVG96] J.C. Bolot, A. Vega García, “*The Case for FEC-based Error Control for Packet Audio in the Internet*”, ACM Multimedia Systems, 1996.
- [Codecs] “*Audio codecs*”, [http ://www.cs.columbia.edu/~hgs/audio/codecs.html](http://www.cs.columbia.edu/~hgs/audio/codecs.html)
- [Cox97] R. Cox, “*Three New Speech Coders from the ITU Cover a Range of Applications*”, IEEE Communications Magazine, pp. 40-47, Sep. 1997.
- [DLW93] B. J. Depsey, J. Liebeherr, A. C. Weaver, “*A New Error Control Scheme for Packetized Voice over High-Speed Area Networks*”, Computer Science Department, University of Virginia, U. S. A., 24 May 1993.
- [Gar96] A. Vega García, “*Mécanismes de contrôle pour la transmission de l’audio sur l’Internet*”, thèse de doctorat, 30 Oct. 1996.
- [RGP98] M. Rocchetti, V. Ghini, G. Pau, P. Salomoni, M. E. Bonfigli, “*Design and Experimental Evaluation of an Adaptive Playout Delay Control Mechanism for Packetized Audio for use over the Internet*”, Technical Report UBLCS-98-4, Department of Computer Science, University of Bologna, Nov. 1998.
- [Sem99] Javier Gozálvéz Sempere, “*An Overview of the GSM System*”, Communications Division Department of Electronic&Electrical Engineering, University of Strathclyde Glasgow, Scotland 1999. [http ://www.comms.eee.strath.ac.uk/~gozalvez/gsm/gsm.html](http://www.comms.eee.strath.ac.uk/~gozalvez/gsm/gsm.html)
- [Sch92] H. Schulzrinne, “*Voice Communication Across the Internet : A Network Voice Terminal*”, Research Report, Department of Electrical and Computer Engineering, University of Massachusetts at Amherst, Jul. 1992.
- [HAH95] V. Hardman, M. Angela, M. Handley, A. Watson, “*Reliable Audio for Use over the Internet*”, Proc. INET’95.
- [ITU88] ITU Rec. G.711, “*Pulse Code Modulation (PCM) of Voice Frequencies*”, 1988.

- [ITU90] ITU Rec. G.726, “*40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)*”, Jun. 1990.
- [ITU96a] ITU Rec. G.723.1, “*Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s*”, Mar. 1996.
- [ITU96b] ITU Rec. G.729, “*Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*”, Mar. 1996.
- [ITU96c] ITU Rec. G.729 Annex A, “*Reduced Complexity 8 kbit/s CS-ACELP Speech Codec*”, Nov. 1996.
- [Jay81] N. S. Jayant, S. W. Christensen. “*Effects of Packet Losses in Waveform Coded Speech and Improvements Due to Odd-Even Sample-Interpolation Procedure*”, IEEE Transactions on Communications, Feb. 1981.
- [Jia99] W. Jiang, H. Schulzrinne, “*QoS Measurement of Internet Real-Time Multimedia Services*”, Technical Report CUCS-015-99, Departement of Computer Science, Columbia University, Dec. 1999.
- [KBS98] T. J. Kostas, M. Borella, I. Sidhu, G. M. Schuster, J. Grabiec et J. Mahler, “*Real-Time Voice Over Packet-Switched Networks*”, IEEE Network, Jan./Fev. 1998.
- [Kle67] E. Klemmer, “*Subjective Evaluation of Transmission Delay in Telephone Conversations*”, Bell System Technical Journal, Jul. 1967.
- [KSB98] T. Kostas, M. S. Borella, “*Analysis of internet delay and loss measurements*”, Network Interop, 1998.
- [RAT] “*RAT - Robust-Audio Tool*”, Computer Science, University College London, London, UK, <http://www.mice.cs.ucl.ac.uk/multimedia/software/rat/>
- [RFC768] J. Postel, “*User Datagram Protocol (UDP) specification*”, RFC 768, Aug. 1980.
- [RFC793] J. Postel, “*Transmission Control Protocol (TCP) specification*”, RFC 793, Sep. 1981.
- [RFC1305] David L. Mills, “*Network Time Protocol (Version 3), Specification, Implementation and Analysis*”, RFC 1305, Mar. 1992
- [RFC1889] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, “*RTP : A Transport Protocol for Real-Time Applications*”, RFC 1889, Jan. 1996.
- [RFC2508] S. Casner, V. Jacobson, “*Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*”, RFC 2508, Feb. 1999.
- [RFC1890] H. Schulzrinne, “*RTP Profile for Audio and Video Conferences with Minimal Control*”, RFC 1890, Jan. 1996.
- [RGP98] M. Rocchetti, V. Ghini, G. Pau, P. Salomoni, M. E. Bonfigli, “*Design and Experimental Evaluation of an Adaptive Playout Delay Control Mechanism for Packetized Audio for use over the Internet*”, Technical Report UBLCS-98-4, Departement of Computer Science, University of Bologna, May 1998.
- [SCK00] H. Sanneck, G. Carle, R. Koodli, “*A frame model for packet loss metrics based on loss run lengths*”, SPIE/ACM SIGMM Multimedia Computing and Networking Conference, Jan. 2000.
- [Sch92] H. Schulzrinne, “*Voice Communication Across the Internet : A Network Voice Terminal*”, Technical report, University of Massachusetts, Jul. 1992.

- [Sus00] J. F. Susbielle, “*Internet multimédia et temps réel*”, Editions Eyrolles, 2000.
- [Tools] “*RTP Tools*”, [http ://www.cs.columbia.edu/IRT/software/rtptools/](http://www.cs.columbia.edu/IRT/software/rtptools/)