HAL
open science

Distribution de contenus multimédia par fot continu : un état de l'art

Jean-Christophe Mignot

To cite this version:

Jean-Christophe Mignot. Distribution de contenus multimédia par fot continu : un état de l'art. [Rapport de recherche] 2000-40, CNRS; ENS Lyon; INRIA Grenoble - Rhône-Alpes; UCBL. 2000. hal-01293181

HAL Id: hal-01293181

https://hal.science/hal-01293181

Submitted on 24 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Laboratoire de l'Informatique du Parallélisme

École Normale Supérieure de Lyon
Unité Mixte de Recherche CNRS-INRIA-ENS LYON n° 5668

Distribution de contenus multimédia par flot continu : un état de l'art

Jean-Christophe Mignot

Novembre 2000

Research Report N° RR2000-40

École Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33(0)4.72.72.80.37
Télécopieur : +33(0)4.72.72.80.80
Adresse électronique : `lip@ens-lyon.fr`

Distribution de contenus multimédia par flot continu : un état de l'art

Jean-Christophe Mignot

Novembre 2000

Abstract

This paper presents a survey of the state-of-the-art techniques and prototypes for streaming over the Internet. The basic principles are presented, the most important approaches (RTP, RTSP) are described.
This work was partially supported by the French Ministry of Industry within the CHARM Project, contract n° 99.2.93.0116

Keywords: Real-time, streaming, Internet protocols

Résumé

Ce document présente un état de l'art des protocoles et outils de streaming disponibles actuellement. Les principes de base sont décrits, les principaux travaux actuels (RTP, RTSP) sont présentés.
Ce travail a été partiellement financé par le ministère de l'Industrie dans le cadre du contrat CHARM, convention n° 99.2.93.0116

Mots-clés: temps réel, diffusion par flot continu, protocoles Internet

1 Introduction

Avec l'explosion de l'Internet et l'apparition des ordinateurs personnels dans le grand public, des objets multimédias sont de plus en plus diffusés. Tout un chacun veut pouvoir diffuser un petit texte, la photo du petit dernier, les miaulements du chat ou la vidéo des premiers pas du plus grand. Pendant longtemps, le mode de diffusion de ces médias a toujours été le même : le fichier est d'abord téléchargé puis visionné. Depuis quelques années, sont apparues des techniques permettant de diffuser les sons et vidéos de manière plus appropriée. Au lieu d'être d'abord téléchargé puis visionné, le fichier contenant le son ou la vidéo est affiché par flot : les données sont traitées au fur et à mesure de leur arrivée, en temps réel. Par exemple, pour une vidéo, les paquets de données reçus de l'Internet seront immédiatement décompressés et les images affichées au fur et à mesure de leurs arrivées. Cette technique est appelée dans la littérature anglophone (et souvent dans la littérature francophone et dans la suite de ce rapport) technique de streaming. Nous parlerons dans la suite de ce rapport de vidéo ou de son streamé pour qualifier une vidéo ou un son distribué de cette manière. L'utilisateur n'a plus à patienter pendant le chargement initial du fichier ; en revanche, la bande passante du réseau depuis le serveur d'origine jusqu'à l'utilisateur doit être supérieure ou égale à la bande passante nécessaire pour le visionnage de la vidéo.

Or, ces bandes passantes sont souvent très élevées. Par exemple, il est possible qu'un fichier contenant un fichier encodé au format AVI non compressé contienne 1Mo ou davantage de données pour une seconde de vidéo. De tels fichiers nécessitent non seulement des capacités de stockage importantes mais aussi des unités de stockage capable de fournir les données avec une bande passante aussi élevée. A titre d'exemple, la bande passante d'une CD-ROM est de 150Ko/s, celle d'un disque dur standard de 5Mo/s. C'est une des raisons pour lesquellee les objets multimédia envoyés sur l'Internet sont compressés. Plusieurs types de compression sont disponibles pour les différents types de médias. Concernant les données audio, citons les formats `au` le format défini par `SUN`, `wav` le format de `MicroSoft` et le fameux standard `MP3`. Pour les vidéos on peut citer `avi` le format défini par `MicroSoft`, `qt` défini par `Apple`, `ram` défini par `Real Networks` et les standards `MPEG1`, `MPEG2` et `MPEG4`.

Les textes et les images sont des données statiques. Pour ces données, s'il est souhaitable que les temps de propagation du serveur jusqu'au client soient les plus courts possibles, il est cependant tolérable d'attendre un certain délai (jusqu'à 10s). Dans la suite de ce document, nous nous intéresserons aux applications dont les données peuvent contenir des sons ou des vidéos streamées. Ces applications sont typiquement très sensibles aux délais de propagation. Suivant l'application considérée, les paquets qui ne sont pas arrivés dans un certain délai (de 100ms à 5s) sont devenus inutiles. Cependant, ce type d'application est souvent tolérant aux pannes : perdre une image dans une vidéo n'est pas grave et un paquet perdu dans un flux audio créera tout au plus un crachement. En terme de ressources en service, les applications multimédia sont donc diamétralement opposées aux applications à données statiques : les applications multimédia sont sensibles aux délais et tolérantes aux pannes, les applications à contenu statique sont tolérantes aux délais et intolérantes aux pannes.

On distingue classiquement trois grands types d'application multimédia :

- Diffusion par flot continu de fichiers audio ou vidéo. Pour ces applications, les clients demandent des sons ou vidéos aux serveurs. Les sons peuvent être des cours, des chansons etc. Les vidéos peuvent être des cours, des clips, des films entiers... Après un court délai de latence, la diffusion commence. L'utilisateur a la possibilité d'arrêter la diffusion en cours, de la reprendre, retourner en arrière, avancer en mode accéléré, exactement comme avec un magnétoscope. La littérature anglophone désigne souvent ce type de diffusion par l'expression VOD pour Video-On-Demand.
- Diffusion partielle par flot de données audio ou vidéo en direct (live dans la littérature anglophone).
- Audio et/ou vidéo interactive en temps réel (visioconférence).

Excepté en réseau local, le réseau Internet n'assure pas une qualité de service suffisante au multimédia sur les communications qu'il établit. La couche réseau du protocole Internet, IP, fournit un service de type "best-effort". L'Internet fait de son mieux pour déplacer chaque paquet de données depuis la source jusqu'à la destination le plus vite possible. Cependant, aucune garantie n'est fournie sur le délai de transmission d'un point à un autre, sur la variabilité des délais de transmission des paquets à l'intérieur d'un même flot de données ("packet jitter") et même sur la perte éventuelle de paquet. Le délai de transmission, s'il devient trop long peut conduire à la perte de paquets pour les applications qui ne peuvent pas attendre la diffusion

d'un paquet donné (son ou vidéo). La variation des délais, si le visionneur joue les paquets dès leur arrivée, peut rapidement rendre un flux inexploitable. Enfin, la perte de paquets peut rendre quasi exsangue toute transmission. Cependant, des techniques adaptées permettent de réduire l'influence des dégradations de ces paramètres tant qu'elles restent dans des limites raisonnables. Les paquets perdus peuvent être ignorés ou interpolés. L'influence des variations de délais peut être limitée par l'utilisation de numéros de séquence, d'une estampille temporelle et d'un délai de visionnage. Ces données sont incluses dans le protocole de communication RTP décrit dans la section 2 de ce document.

RTP et RTSP désignent des protocoles mis au point pour permettre la diffusion par flot sur Internet. Les trois plus grands visionneurs de vidéos streamées actuels, le RealPlayer de Real Networks (`http://www.real.com`), le MediaPlayer de Microsoft (`http://www.microsoft.com`) et QuickTime 4 de Apple (`http://www.apple.com`), permettent tous l'utilisation de ces protocoles du domaine public.

RTP (Real Time Protocol, voir [10] et la très documentée `http://www.cs.columbia.edu/~hgs/rtp`) est un protocole qui fournit des fonctions pour le transport sur un réseau point-à-point de données temps-réel, tels le son et la vidéo, avec des services de diffusion point-à-point ou de diffusion partielle. RTP ne gère pas la réservation de ressources et ne garantit pas de qualité de service temps-réel. Le transport des données est augmenté d'un protocole de contrôle (RTCP, Real Time Control Protocol) pour permettre le monitoring de manière extensible des données diffusées sur un grand réseau multicast et pour fournir des fonctionnalités minimales de contrôle et d'identification. RTP et RTCP ont été définis de manière indépendante des couches de transport et réseau sous-jacentes.

RTSP (Real Time Streaming Protocol, voir [11], et la très documentée `http://www.cs.columbia.edu/~hgs/rtsp`) est un protocole de niveau applicatif pour le contrôle de la diffusion de données temps-réel. Il fournit un cadre extensible pour permettre le contrôle et la diffusion à la demande de données temps-réel comme la vidéo et le son. L'origine des données peut être un enregistrement en direct (présentation live) ou un fichier préalablement enregistré. Ce protocole autorise le contrôle de plusieurs sessions de diffusion de données, propose des moyens pour choisir des canaux de diffusion tels UDP, multicast UDP ou TCP, et fournit des moyens pour choisir des mécanismes de diffusion basés sur RTP.

Dans la suite de ce document, nous présenterons en section 2 les protocoles RTP et RTSP permettant la diffusion par flot d'objets multimédia. La section 3 sera dévolue au protocole de contrôle du transfert des données, RTSP.

2 Le Real Time Protocol, RTP et RTCP

Les briques de base pour le transport temps-réel d'information sont :

- RTP, le real time transport protocol
- RTCP, le real time transport control protocol
- des profils définissant les formats des données utiles pour les différents types de flux.

Pour résoudre les problèmes liés à la diffusion de données vidéo/audio, il est nécessaire d'ajouter un en-tête aux blocs de données transportés spécifiant un numéro de séquence, une estampille temporelle et une durée de présentation. RTP est un standard proposant une structure pour de tels paquets de données. Il inclut aussi en en-tête d'autres informations utiles. Il permet de transporter des formats standards tels Wav ou GSM pour les sons, MPEG1 et MPEG2 pour la vidéo.

RTCP a aussi été défini dans le RFC1889 (`http://www.cis.ohio-state.edu/htbin/rfc/rfc1889.html`) et peut être utilisé en conjonction avec RTP. Ce protocole standardise la nature des statistiques échangées entre émetteurs et récepteurs lors de session RTP incluant de nombreux receveurs et un ou plusieurs émetteurs. Son rôle est d'assurer le contrôle de la qualité de la session en cours.

RTP est un représentant d'un nouveau style de protocoles, suivant les principes enoncés dans [4], d'empaquetage au niveau applicatif (application level framing) et de traitement intégré des couches (integrated layer processing). Ainsi RTP est supposé être malléable afin de fournir les informations nécessaires à une application particulière et sera souvent intégré à l'application plutôt qu'implémenté comme couche logicielle. RTP est une spécification de protocole incomplète. Seules sont documentées par les auteurs les fonctions qui sont censées être nécessaires à toutes les applications susceptibles d'utiliser RTP. Par opposition aux protocoles classiques, pour lesquels l'ajout de fonctionnalités passe soit par une généralisation du protocole,

soit par l'ajout d'options, RTP est prévu pour être adapté via des modifications et/ou des additions aux entêtes. Donc, une spécification complète de RTP pour une application donnée nécessite un ou plusieurs documents adjoints :

- un document de spécification de profil qui définit l'association entre un ensemble de codes de types de données utiles et les formats associés ;
- un document de spécification de format de données utiles qui définit comment les données vidéo/audio seront transportées dans RTP.

De nombreuses applications ont été écrites en s'appuyant sur les spécifications de RTP. Ces applications concernent des outils de conférence audio et/ou vidéo tout autant que des outils de diagnostic comme le monitoring du trafic. Il est à noter que les spécifications de RTP n'abordent pas les problèmes de consommation de bande passante; Par conséquent, une mauvaise utilisation peut engendrer une consommation excessive et perturber la qualité de service du réseau. Il est donc conseillé aux auteurs d'applications d'être prudents quant à la consommation de ressource Internet et d'indiquer clairement dans leurs documentations les limitations et impacts éventuels de leurs implémentations.

2.1 RTP

Typiquement, RTP est exécuté au dessus de UDP (voir Fig 1). Les blocs de données, générés par la sortie de l'application multimédia, sont encapsulés dans des paquets RTP, puis chaque paquet RTP est à son tour encapsulé dans un segment UDP. Considérons l'exemple d'une application téléphonique sur RTP. Supposons que la source soit encodée au format PCM à 64 kb/s et que les échantillons soient disponibles toutes les 20 ms. L'application serveur ajoute à chaque bloc un entête RTP qui inclut le type de codage, un numéro de séquence et une estampille de temps. L'en-tête et le bloc de données forment le paquet RTP. Ce paquet est ensuite envoyé sur un socket UDP où il sera empaqueté dans un paquet UDP. Du coté client, c'est un paquet RTP qui est reçu via le socket. L'application cliente peut extraire le bloc de données audio et utiliser l'en-tête pour le décoder puis le jouer correctement.

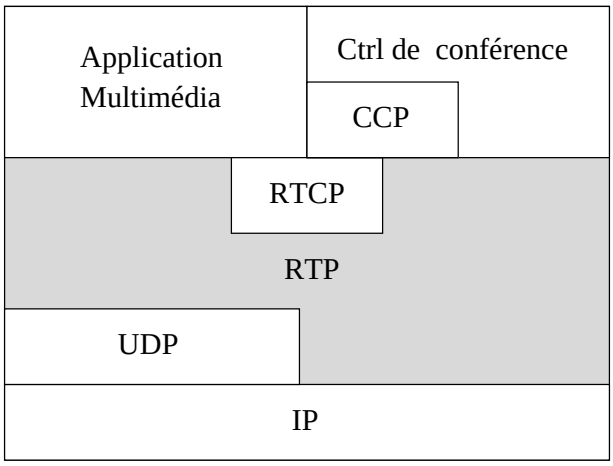

FIG. 1: La place de RTP dans les couches réseau

RTP permet d'associer un flot de paquets à chaque type de source, par exemple un flot pour un micro et un autre pour une caméra. Dans ce cas, pour une vidéoconférence à deux, 4 flots seront créés, 2 flots audio et 2 flots vidéo dans chaque sens. Cependant dans de nombreux formats de compression dont MPEG1 et MPEG2, les images et le son sont mélangés. Pour de tels cas, seul un flot est créé dans chaque sens. RTP est notablement compatible avec les réseaux multicast. Les paquets peuvent être envoyés sur des arbres de multi-diffusion un-vers-tous et tous-vers-tous.

2.1.1 Mixeurs et translateurs

Nous avons supposé jusqu'ici que tous les sites de la session RTP voulaient recevoir les données dans le même format. Or, ce n'est pas toujours le cas. Considérons le cas où certains des membres de la session RTP sont connectés via un liaison bas débit alors que les autres bénéficient d'une connexion à haut-débit. Plutôt que d'imposer à tous les connectés les mêmes formats bas débit, un dispositif relai pour RTP, appelé mixeur, peut être disposé près du réseau à faible bande passante. Le mixeur re-synchronize les paquets reçus de la partie haut débit, mixe les différents flux en un unique flux, traduit les données codées pour un réseau haut débit en utilisant un code adapté au bas débit et renvoie le paquet sur le lien à faible bande passante. L'entête RTP permet aux mixeurs d'identifier les émetteurs combinés dans le paquet mixé afin de pouvoir indiquer la source correcte aux receveurs.

Définition : un mixeur est un système intermédiaire qui reçoit des paquets RTP depuis une source, change éventuellement le format des données, combine d'une manière ou d'une autre les paquets et envoie un nouveau paquet RTP. Les multiples sources n'étant pas a priori synchronisées, le mixeur fera des ajustements temporels entre les flux et générera son propre temps pour le flux combiné. Les paquets sortant du mixeur auront donc tous le mixeur pour source de synchronisation.

Certains participants d'une session RTP peuvent ne pas être joignables en multicast, si, par exemple, ils sont derrière une barrière coupe-feu qui ne laissera passer aucun paquet IP. Pour ces cas, un autre type de relais RTP sera utilisé, appelé translateur. Deux translateurs seront installés, un de chaque coté du coupe-feu. Celui à l'extérieur du réseau sécurisé collectera les paquets multicast puis les transmettra via une liaison sécurisée au translateur situé à l'intérieur du réseau. Ce translateur renvoie alors ces paquets, sous forme multicast à un groupe multicast restreint à l'intérieur du réseau.

Définition : un translateur est un système intermédiaire qui renvoie des paquets RTP en laissant leur identificateur de source de synchronisation intact. Parmi les translateurs, on trouve les dispositifs qui convertissent les codages sans multiplexer, les réplicateurs de multicast vers unicast et les filtres logiciels pour coupe-feu.

2.1.2 Les entêtes RTP

Un entête RTP est représenté par la figure 2. Les 12 premiers octets sont toujours présents dans tout entête RTP tandis que les identificateurs des CSRC ne sont présents que lorsqu'un mixer (voir plus loin) est utilisé.

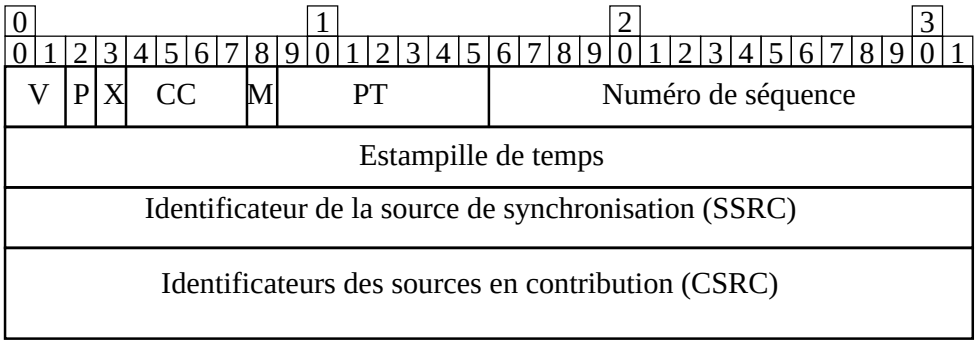

FIG. 2: Un entête RTP

Les différents champs sont :

- **V**, le numéro de version de RTP sur 2 bits. La version actuelle est 2.
- **P**, padding. Si ce bit est à 1, il y a un ou plusieurs octets à la fin du paquet qui ne font pas partie des données utiles. Le dernier de ces octets représente le nombre d'octet à ignorer. Cette facilité permet par exemple d'insérer des clés pour le cryptage des données ou encore d'insérer d'autres paquets RTP.
- **X**, extension. Si ce bit est à 1, l'en-tête de taille fixe est suivi par une extension d'en-tête de format défini plus loin dans ce document.

- **CC**, CSRC count, sur 4 bits. Ce champ représente le nombre d'identificateurs des sources en contribution qui sont présents en fin de l'entête.
- **M**, marqueur, 1 bit. L'interprétation de ce marqueur dépend du profil. Le but est de permettre la signalisation événements importants comme la fin d'une image d'une vidéo par exemple. Un profil permet de définir d'autres marqueurs en changeant le nombre de bits dans le champ de la définition du type des données utiles (voir plus loin).
- **PT**, payload type ou type de données utiles, sur 7 bits. Ce champ identifie le format des données utiles transportées et détermine son interprétation par l'application. Il existe une table fixe faisant correspondre les codes de type de données utiles aux formats de données. D'autres codes de type de données utiles peuvent être définis dynamiquement en passant par des mécanismes non-RTP. La liste des codes et des formats définis peut être trouvée dans l'Internet-Draft [8]. Un serveur RTP émet un unique type de données à un instant donné, ce champ n'est pas censé permettre le multiplexage de flots multimédia différents. Les tableaux 1 et 2 suivants présentent quelques types de données déjà définis.

# de type de données	Format audio	Fréquence d'échantillonnage	Débit
0	PCM mu-law	8 KHz	64 Kbps
3	GSM	8 KHz	13 Kbps
14	MPEG Audio	90 KHz	-

TAB. 1: Quelques types de données audio supportés par RTP

# de type de données	Format video
26	Motion JPEG
31	H.261
32	Video MPEG1
33	Video MPEG2

TAB. 2: Quelques types de données vidéo supportés par RTP

- **Numéro de séquence** sur 16 bits. Le numéro de séquence est incrémenté à chaque nouveau paquet RTP émis et peut être utilisé par le client pour identifier et traiter les pertes de paquets. Pour des raisons de sécurité, la valeur initiale de ce numéro est aléatoire afin d'éviter un décryptage trop facile des flots.
- **Estampille de temps**, sur 32 bits. Ce champ contient le temps d'échantillonnage du premier octet du paquet. Il est utilisé pour les calculs de synchronisation et de variations de délai de livraison (jitter). L'estampille est calculée à partir de l'horloge d'échantillonnage sur le serveur (pas l'horloge centrale). Par exemple, pour du son, l'horloge de l'estampille sera incrémentée de 1 à chaque nouvelle période d'échantillonnage (par exemple toutes les 250 (sec pour un échantillonnage à 4kHz). Si l'application fournit des paquets de 100 échantillons, l'estampille est incrémentée de 100 à chaque nouveau paquet RTP. L'horloge d'estampillage doit rester active à un rythme constant même si la source est inactive (silence). Le type de données utiles définit la fréquence d'échantillonnage pour les types prédéterminés et devrait définir la fréquence d'échantillonnage pour les types définis dynamiquement. Comme pour le numéro de séquence, la première estampille de temps est aléatoire. Plusieurs paquets RTP peuvent avoir la même estampille, par exemple s'ils appartiennent à la même image d'une vidéo. Des paquets consécutifs peuvent avoir des estampilles non monotones si les données ne sont pas transmises dans l'ordre où elles ont été échantillonnées, comme c'est le cas pour les frames vidéo interpolées de MPEG, mais le numéro de séquence, lui, restera monotone.
- **Identificateur de la source de synchronisation**, ou SSRC pour Synchronization Source, sur 32 bits. Ce champ identifie la source de synchronisation, i.e. l'émetteur du flot. Typiquement, chaque flot

dans une session RTP a un identificateur unique. Cet identificateur est choisi aléatoirement pour éviter que deux sources de synchronisation d'une même session RTP n'aient le même numéro. Un algorithme pour le choix de ce numéro est fourni dans [10]. Même s'il garantit que la probabilité que deux sources aient le même numéro est faible, les auteurs préviennent les développeurs de tenir compte d'éventuelles collisions dans les applications et un algorithme de résolution est proposé.

- **Identificateurs des sources en contribution**, ou CSRC pour Contribution Sources, sur 32 bits. Cette liste identifie les sources en contribution pour les données transportées dans le paquet. Le nombre d'identificateurs est défini par le champ CC de l'en-tête. Si le nombre de sources est supérieur à 15, seules 15 seront identifiées. Les CSRC sont insérés par les mixeurs en utilisant les SSRC des différentes sources. Par exemple, pour le son, les SSRC de toutes les sources qui ont été mélangées par le mixeur pour créer le paquet seront listées, permettant au receveur d'identifier les différents intervenants.

2.1.3 Multiplexage de flots RTP

RTP ne permet pas le multiplexage de données à l'intérieur d'un flot. La possibilité de multiplexage est fournie par les adresses de destination de transport (adresse réseau *et* numéro de port) qui définissent la session RTP. Par exemple, pour une vidéoconférence audio/vidéo, un flot sera créé pour chaque type de données et chaque flot sera dirigé vers une adresse de transport (machine et numéro de port) distincte. Le multiplexage, soit par le SSRC, soit par le PT, n'est pas pris en compte par RTP pour des raisons de faisabilité (voir [10] pour plus de détails).

2.1.4 Modifications des entêtes RTP par profile

Bien que RTP ait été défini pour remplir toutes les fonctionnalités nécessaires, des mécanismes permettent de modifier ou d'enrichir les entêtes des paquets RTP tout en restant compatible avec les outils d'enregistrements et de monitoring existants :

- Le bit **M** de l'entête et le type de données utiles, PT, contiennent des informations interprétées grâce au profil mais sont quand même attribués dans les en-têtes fixes du protocole car ils sont supposés être utilisés par de nombreuses applications. L'octet contenant ces champs peut être redéfini par le profil par exemple pour utiliser plus de bits de marquage.
- Des informations supplémentaires peuvent être stockées dans les données utiles, par exemple en insérant un entête supplémentaire au début de chaque section de données utiles.
- Des données peuvent être insérées juste après le champ SSRC. Les applications pourront avoir ainsi un accès rapide à ces informations et les outils de monitoring et d'enregistrement pourront toujours opérer en ne considérant que les 12 premiers octets.

Extension de l'entête Un dispositif est fourni permettant de rajouter des informations dans les en-têtes des paquets de données. Ce mécanisme a été fait de manière à ne pas perturber les applications ignorant l'extension de l'en-tête. Les auteurs insistent sur l'aspect exceptionnel de l'utilisation de ce mécanisme et préconisent plutôt la méthode précédente quand c'est possible.

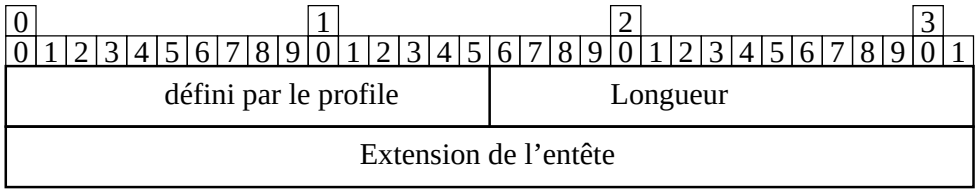

FIG. 3: L'entête d'extension d'entête RTP

Si le bit **X** de l'entête est à 1, alors une extension d'en-tête de longueur variable est insérée à la fin de l'entête juste après la liste des CSRC. L'extension d'en-tête (voir Fig. 3) contient un champ de 16 bits définissant le nombre de mots de 32 bits contenus dans l'extension (en excluant les 4 octets de l'entête d'extension).

2.2 RTCP

Le protocole RTCP permet de contrôler en temps réel l'état d'une session RTP typiquement représentée par la figure 4. Les paquets RTCP ne comportent pas de données multimédia mais transportent des rapports qui contiennent des statistiques utiles à l'application. Parmi ces statistiques, on trouve le nombre de paquets envoyés, le nombre de paquets perdus et la variation des délais (jitter). Les spécifications de RTP ne définissent pas la manière d'utiliser ces informations, leur utilisation est laissée au libre choix des développeurs. Elles peuvent servir par exemple à modifier les taux de compression en cours de session. Les données RTCP sont transportées en utilisant le même mécanisme de distribution que celui des données. Le protocole utilisé doit donc permettre le multiplexage des paquets de données et de contrôle, par exemple, avec RTP, en utilisant deux ports UDP distincts.

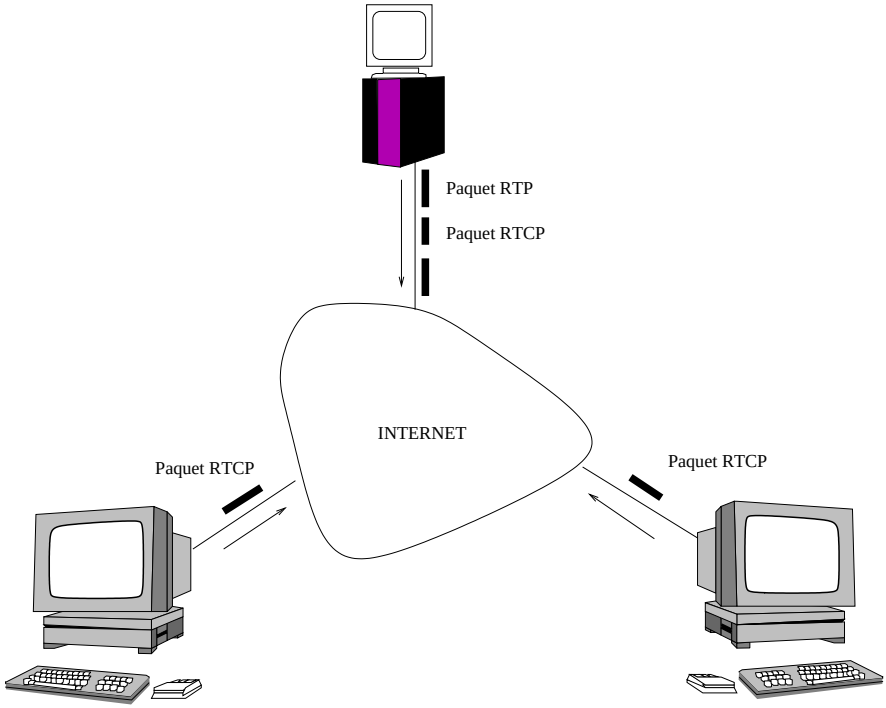

FIG. 4: Une session RTP typique

Le protocole RTCP assure quatre fonctions principales :

- La première des fonctions est d'assurer un feed-back sur la qualité de la distribution des données. Cette information peut être utilisée pour le contrôle des codages adaptatifs (changements de taux de compression) mais aussi pour identifier des problèmes dans la distribution. Le fait que les rapports de réception soient envoyés à tous les participants permet à un participant connaissant des problèmes de déterminer si le problème est local ou global. De la même manière, ce feed-back permet de surveiller le réseau pour des acteurs n'appartenant pas forcement à la vidéoconférence comme par exemple les prestataires de services.
- RTCP fournit pour les sources de flux un identificateur persistant, de niveau transport, appelé `CNAME` pour canonical name. Il est à noter que si le SSRC d'un flux peut changer (suite à un conflit d'identificateur ou à un redémarrage du programme) le CNAME reste identique. Le CNAME sert aussi à associer différents flux émanant d'un même émetteur, par exemple pour synchroniser des flux audio et vidéo.
- Puisque tous les participants envoient leur rapports à tous les autres, les problèmes d'extensibilité ont été abordés. Ainsi, chaque participant évalue le nombre de participants et adapte la fréquence d'envoi des rapports de manière à ne pas engorger le réseau.

– RTCP fournit aussi des informations nécessaires pour un contrôle minimal des sessions comme par exemple l'identifiant d'un participant pour l'afficher dans l'interface utilisateur. RTCP peut ainsi être un moyen pratique de transmettre des informations à tous les participants mais il n'est pas nécessaire que tout passe par lui.

2.2.1 Les paquets RTCP

Les spécifications contenues dans [10] définissent 5 types de paquets RTCP : les rapports d'émetteur (SR pour Sender Report), de récepteur (RR pour Receiver Report), les descriptions de source (SDES pour Source Description), les paquets "BYE" et des paquets de fonctions spécifiques aux applications (APP).

Les paquets RTCP commencent tous par une partie fixe, semblable à celle des paquets RTP, suivie par d'éléments structurés de taille éventuellement variable, suivant le type de paquet, mais toujours alignés sur 32 bits. Un tel datagramme est représenté par la figure 5 où # représente soit un SSRC soit un CSRC et où `R` est un nombre aléatoire sur 32 bits en cas de session cryptée. L'alignement et un champ de longueur permettent de rendre les paquets concaténables aisément pour les envoyer dans un seul paquet du protocole sous-jacent, par exemple UDP.

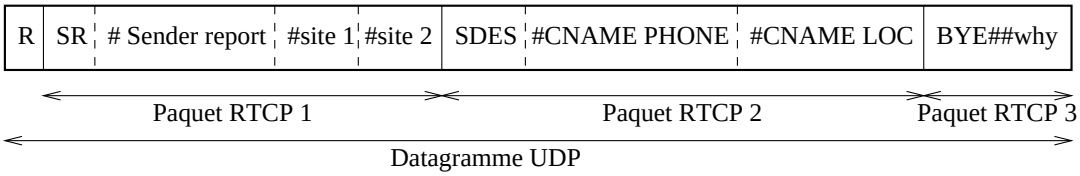

FIG. 5: Paquets RTCP concaténés dans un datagramme UDP

Des contraintes ont été définies pour les paquets composés :

– Les paquets de type SR et RR doivent être envoyés aussi fréquemment que les contraintes sur la bande passante le permettent afin d'obtenir de bonnes statistiques. Chaque paquet RTCP composé doit donc contenir un paquet SR ou RR.

– Les nouveaux participants à une session doivent recevoir aussi rapidement que possible le CNAME pour une source. Chaque paquet RTCP composé doit donc contenir un paquet SDES.

Dans la suite de cette section, nous décrirons succintement les différents paquets RTCP. Le lecteur interessé pourra se reporter à [10] pour une description complète.

Les paquets de réception Les paquets de réception sont émis par les receveurs d'un flot données. Pour chaque flot de données reçu, un receveur génère un rapport de réception. Il agrège ces rapports en un unique paquet RTCP. Le paquet est ensuite envoyé en multicast à tous les membres de la session. Les informations les plus importantes sont :

– la source de synchronisation (SSRC) du flot RTP correspondant au rapport ;

– le nombre des paquets perdus dans le flot. Chaque receveur pourra ainsi calculer le ratio des paquets perdus et éventuellement passer à un taux de compression plus élevé afin de réduire l'encombrement du réseau ;

– le dernier numéro de séquence reçu dans le flot de paquets RTP ;

– la variation de délai des paquets (jitter).

Les paquets d'émission Pour chaque flot, l'émetteur génère des rapports d'émission qui contiennent :

– la source de synchronisation du flot RTP ;

– l'estampille de temps et l'heure du dernier paquet RTP généré pour le flot ;

– le nombre de paquets envoyés dans le flot ;

– le nombre d'octets envoyés dans le flot.

Les rapports des émetteurs peuvent être utilisés pour synchroniser différents flux. Par exemple dans le cas d'une vidéoconférence vidéo/audio, les estampilles de temps portées par les paquets RTP des flux audio et vidéo ne sont liées qu'aux fréquences d'échantillonnage et non pas à l'heure (i.e. le "vrai" temps). Le rapport contenant l'estampille de temps et l'heure du dernier paquet généré, les receveurs peuvent aisément synchroniser les deux flux.

Les paquets de description de source Pour chaque flot émis, l'émetteur génère des paquets de description de source. Ces paquets contiennent des informations relatives à la source du flot tels l'adresse e-mail de l'envoyeur, son nom et l'application qui a généré les paquets RTP. Le SSRC est aussi inclus. Ces paquets fournissent la correspondance entre l'identificateur de source et les noms d'hôte et d'utilisateur.

Les paquets "BYE" Les paquets "BYE" indiquent qu'une ou plusieurs sources ne sont plus actives. Si un paquet "BYE" est reçu par un mixeur, il est renvoyé intact. Si un mixeur s'arrête, il devrait envoyer un paquet "BYE" listant toutes les sources qu'il gère ainsi que son propre identificateur de SSRC.

Les paquets spécifiques aux applications Ce type de paquet a été défini à des fins expérimentales pour permettre de tester de nouvelles applications. Après les tests et si ce type de paquets est susceptible d'intéresser d'autres personnes que les auteurs, il est préconisé de faire enregistrer le type auprès de l'Internet Assigned Numbers Authority.

2.2.2 La bande passante consommée par RTCP

Le lecteur attentif aura remarqué que RTCP, tel qu'exprimé jusqu'à maintenant, a de potentiels problèmes d'extensibilité. En effet, considérons le cas d'un unique émetteur pour un grand nombre de receveurs. Si les receveurs émettent régulièrement des paquets RTCP, la bande passante consommée par ces paquets peut devenir largement supérieure à celle consommée par l'émetteur. Notons que si la quantité de trafic RTP envoyé sur le réseau multicast n'augmente pas avec le nombre de receveurs, celle du trafic RTCP augmente linéairement avec le nombre de receveurs. Pour pallier ce problème, RTCP modifie la fréquence d'envoi des paquets RTCP en fonction du nombre de participants. RTCP tente de limiter sa consommation à 5% de la bande passante de la session.

3 Le Real Time Streaming Protocol, RTSP

RTSP (voir [11], `http://www.realnetworks.com/devzone/library/rtsp`) est un protocole applicatif pour contrôler le transfert de données temps-réel mis au point conjointement par l'Université de Colombia, Real Networks et Netscape Corp. Il fournit un cadre extensible pour une livraison à la demande, contrôlée, de données temps-réel telles la vidéo/audio. Les sources des données peuvent être des flux live (spectacle en direct) ou enregistrés. Ce protocole est étudié pour contrôler plusieurs sessions de transfert de données temps-réel, fournit les moyens de choisir un canal de communication comme UDP, multicast UDP ou TCP et fournit un moyen de choisir parmi les mécanismes de livraison de RTP. RTSP agit comme une télécommande du serveur de données multimédia via le réseau.

L'ensemble des flots à contrôler est défini par une description de présentation.

Il n'y a pas de notion de connexion RTSP à proprement parler ; à la place, le serveur entretient une session libellée par un identificateur. Une session RTSP n'est en aucun cas liée à un type de connexion de transport tel TCP, au contraire durant une session RTSP, un client peut ouvrir ou fermer plusieurs connections de transport au serveur pour effecteur des requêtes RTSP. Un protocole de transport sans connexion comme UDP peut aussi être utilisé.

Le protocole supporte les opérations suivantes.

- collecte d'un objet multimédia depuis un serveur. Le client peut demander un descripteur de présentation via HTTP ou autre. Si la présentation est multi-diffusée (multicast), le descripteur de présentation contient les adresses et ports à utiliser ;

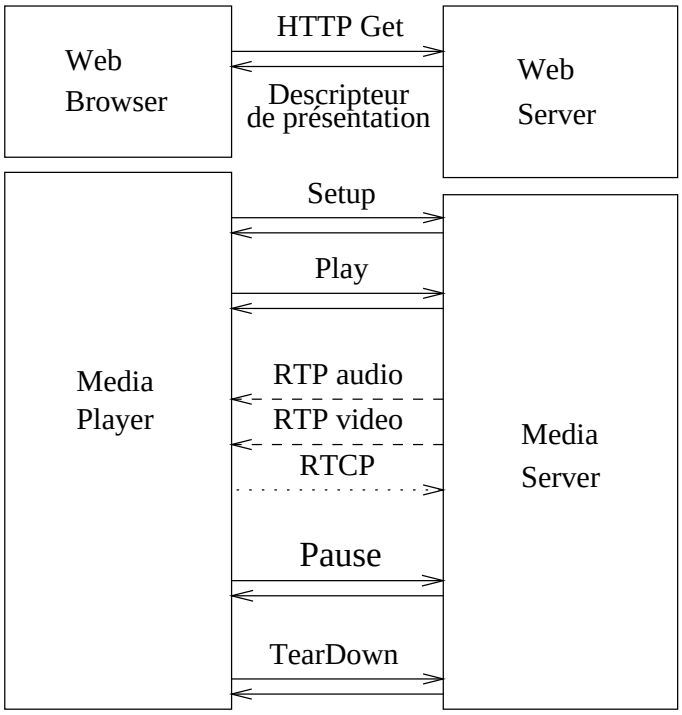

FIG. 6: Interactions entre le client et le serveur avec RTSP

- invitation d'un serveur multimédia à une conférence. Un serveur multimédia peut être invité à une conférence soit pour jouer un objet multimédia, soit pour enregistrer tout ou partie de la présentation ;
- addition d'un objet multimédia dans une présentation. Plus particulièrement pour les présentations en direct, il est utile que le serveur puisse indiquer au client que de nouveaux objets multimédia sont disponibles.

Les requêtes RTSP peuvent être gérées par les proxies, les tunnels et les caches de la même manière que les requêtes HTTP/1.1 ([5]).

Tout comme ftp, RTSP est un protocole "out-of-band". Lors d'une connexion ftp entre deux machines, deux connexions TCP sont créées sur des ports différents, l'une transporte les données, l'autre des informations de contrôle. La connexion de contrôle est dite canal "out-of-band" tandis que l'autre est appelée canal de données.

La figure 6 représente un exemple simple de transactions RTSP. Le navigateur Web demande en premier lieu un descripteur de présentation au serveur Web. Le descripteur de présentation peut contenir une ou plusieurs références à des objets multimédia ainsi que des commandes de synchronisation. Chaque référence à un objet multimédia commence par la méthode de l'URL, rtsp ://. Le fichier descripteur de la présentation est renvoyé au navigateur Web via HTTP en réponse à sa requête. Le navigateur invoque le visionneur *ad hoc* en fonction du champ `content-type :` de la réponse. Le visionneur et le serveur échangent alors une série de messages RTSP. Le visionneur envoie en premier lieu une requête de type "SETUP", à laquelle le serveur répond par une réponse de type "SETUP". Le visionneur envoie une requête "PLAY", par exemple pour de l'audio de basse qualité, le serveur réplique par une réponse "PLAY". Le serveur de flot envoie alors le fichier audio de basse qualité dans le canal de données. Si plus tard, le visionneur envoie une requête "PAUSE", le serveur répond avec la réponse "PAUSE" et le flot de données est stoppé. Pour terminer une session, le visionneur envoie une requête "TEAR DOWN" à laquelle le serveur réplique par une réponse de même type.

La figure 7 représente un descripteur de présentation décrit dans [9]. Dans cette présentation, un flot audio et un flot vidéo sont joués en parallèle dans le groupe synchro. Pour le flot audio, le visionneur de contenu multimédia peut choisir (tag `switch`) entre deux enregistrements, un de haute et l'autre de basse

```
<title>Exemple</title>
<session>
   <group language=fr synchro>
      <switch>
         <track type=audio
                e="PCMU/800/1"
                src="rtsp://audio.tst.com/audio/audio.fr/BasseQual">
         <track type=audio
                e="DVI4/1600/2" pt="90 DVI4/8000/1"
                src="rtsp://audio.tst.com/audio/audio.fr/HauteQuall">
      </switch>
         <track type=video/jpeg"
                src="rtsp://audio.tst.com/video/video1>
   </group>
</session>
```

FIG. 7: Un exemple de descripteur de présentation

qualité.

Chaque session RTSP possède un identificateur de session choisi par le serveur : le client initie la session avec une requête "SETUP", à laquelle le serveur répond avec l'identificateur de la session dans la réponse de type "SETUP". L'identificateur de la session est inclus dans toutes les requêtes RTSP jusqu'à ce que la session soit close par la requête "TEARDOWN".

```
Client: SETUP rtsp://audio.tst.com/twister/audio RTSP/1.0
        Transport: rtp/udp; compression; port=3056; mode=PLAY

Serveur: RTSP/1.0 200 1 OK
         Session 4231

Client: PLAY rtsp://audio.tst.com/twister/audio.en/lofi RTSP/1.0
        Session: 4231
        Range: npt=0-

Client: PAUSE rtsp://audio.tst.com/twister/audio.en/lofi RTSP/1.0
        Session: 4231
        Range: npt=37

Client: TEARDOWN rtsp://audio.tst.com/twister/audio.en/lofi RTSP/1.0
        Session: 4231

Serveur: 200 3 OK
```

FIG. 8: Un exemple de transactions RTSP entre un client et un serveur

La figure 8 montre un exemple de déroulement d'une session RTSP. Dans cet exemple, le visionneur choisit de ne pas jouer la session en entier mais seulement la partie audio basse qualité de la présentation.

RTSP offre de nombreuses autres possibilité qui sortent du cadre de ce document. Le lecteur interessé pourra se reporter à [11] pour une information complète sur RTSP. Il est à noter que les visionneurs les plus répandus actuellement, le Real Player 7 de Real Networks (voir `http://www.realnetworks.com/devzone/library/rtsp/`), le `Windows Media Player` de Microsoft (voir `http://www.microsoft.com/Windows/windowsmedia/en/profiles/NorthropGrumman/`) et `QuickTime 4.0` de Apple (`http://www.apple.com/uk/quicktime/resources/qt4/us/proxy/`), ont tous adopté RTSP.

4 Conclusion

Les protocoles RTP et RTSP peuvent être utilisés pour diffuser par flot des vidéos de type MPEG1/MPEG2. Des profils RTP ont été définis pour l'encapsulation des données audio/vidéo MPEG dans les trames RTP dans [7, 3]. Ces documents décrivent la manière dont seront remplis les champs des en-têtes fixes RTP ainsi que des extensions d'entêtes qui sont utilisés pour transporter des informations spécifiques aux vidéos MPEG1/2 tels les types des images du paquets, l'eventuelle présence d'en-tête de synchronisation du flux MPEG etc. Le lecteur interessé par les formats MPEG pourra se réferrer à [1] pour une information détaillée.

La place de RTP dans le monde de la distribution par flot est prépondérant. Tous les visionneurs disponibles proposent RTP comme moyen de transport. Malheureusement, l'utilisation de ce standard ne garantit pas l'intéropérabilité entre les différents serveurs/visionneurs du marché. La première raison est souvent l'incompatibilité des formats des fichiers streamés. Par exemple, le visionneur Real Player 7 ne reconnaît pas le format le format Quick Time. Mais certains serveurs refusent aussi les connexions demandées par un visionneur autre que celui de la marque du serveur même si le format du fichier est standard comme par exemple MPEG. Notons que les choses pourraient changer rapidement avec la distribution gratuite et en open source de nombreux outils pour la diffusion de vidéos sur le Web dont, par exemple, le serveur de vidéos streamées d'Apple, Darwin.

Références

[1] ISO/IEC International Standard 13818. Generic coding of moving pictures and associated audio information, Nov 1994.

[2] A. Basso, G. L. Cash, and M. R. Civanlar. Transmission of MPEG-2 Streams over Non-Garanteed Quality of Service Networks. In *Proc. of Picture Coding Symposium*, Berlin, Germany, Sept 1997.

[3] M. Reha Civanlar, Glenn L. Cash, and G. Haskel Barry. Rtp Payload Format for Bundled MPEG. Internet Draft, `http://ds1.internic.net/internet-drafts/draft-civanlar.bmpeg-01`, june 1997.

[4] David D. Clark and David L. Tennenhouse. Architectural Considerations for a New Generation of Protocols. In *Proc. of the ACM SIGCOMM'90 Symposium*, sept 1990.

[5] R. Fielding, J. Gettys, J. Mogul, H. Nielsen, and T. Berners-Lee. Hypertext transfer protocol - HTTP/1.1. Internet Draft, RFC 2068, `http://info.internet.isi.edu/in-notes/rfc/files/rfc2068.txt`, Jan 1997.

[6] Fransisco Cortés Gómez and Rheinard Lüling. A Parallel Continuous Media Server for Internet Environments. In *Proc. of High Performance Computing and Networking (HPCN Europe'98)*. Springer, 1998.

[7] Don Hoffman and Vivek Goal. RTP Payload Format for MPEG1/MPEG2 Video. Internet Draft, `http://ds1.internic.net/internet-drafts/draft-ietf-avt-mpeg-00.txt`, june 1995.

[8] J. Reynolds and J. Postel. Assigned Numbers. Internet Draft, RFC 1700, `http://ds1.internic.net/internet-drafts/draft-ietf-avt-profile.txt`, october 1994.

[9] Henning Schulzrinne. A Comprehensive Multimedia Control Architecture for the Internet. In *Proc. of NOSSDAV'97 (Network and Operating System Support for Digital Audio and Video)*, St. Louis, Missouri, May 1997.

[10] Henning Schulzrinne, S. Casner, R. Frederik, and V. Jacobson. RTP : A Transport Protocol for Real-Time Applications. Internet Draft, RFC 1889, `http://www.cis.ohio-state.edu/htbin/rfc/rfc1889.html`, march 1996.

[11] Henning Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). Internet Draft, RFC 2326, `http://info.internet.isi.edu/in-notes/rfc/files/rfc2326.txt`, Avril 1998.