

Procédé de diffusion de contenus en streaming dans un réseau pair à pair

DOMAINE TECHNIQUE GENERAL

La présente invention concerne un procédé de diffusion d'un contenu en streaming dans un réseau pair à pair.

ETAT DE L'ART

Le « streaming » (en français « lecture en continu »), désigne une technique de lecture d'un flux audio ou vidéo « en direct », c'est-à-dire au fur et à mesure qu'il est récupéré depuis Internet par un équipement client. Elle s'oppose ainsi au téléchargement, qui nécessite de récupérer l'ensemble des données du contenu audio ou vidéo avant de pouvoir le lire.

Dans le cas du streaming, le stockage du contenu est provisoire et partiel, les données étant téléchargées en continu dans une mémoire tampon du client (typiquement la mémoire vive), décodées à la volée, envoyées vers une interface de sortie (un écran et/ou des enceintes) puis remplacées par de nouvelles données.

De façon traditionnelle, le contenu est mis à disposition sur un serveur de streaming. Le client souhaitant y accéder envoie une requête pour en récupérer les premiers segments (par segment, on entend un bloc des données du contenu, généralement correspondant à quelques secondes de lecture). Lorsqu'il y a suffisamment de données dans la mémoire tampon (buffer) pour permettre de lire le début du contenu, la lecture démarre. En arrière-plan, le téléchargement du flux se poursuit afin d'alimenter sans cesse la mémoire tampon avec la suite du contenu.

On constate néanmoins que cette approche a ses limites si un grand nombre de clients souhaitent lire le même contenu simultanément : le serveur se retrouve saturé, dans l'incapacité de fournir le contenu à un débit suffisant pour que la lecture soit fluide, et des saccades apparaissent.

Il a récemment été proposé une stratégie alternative basée sur le « peer-to-peer » (P2P, en français pair-à-pair), dans laquelle chaque client agit comme un

serveur pour d'autres clients : on parle de pairs. Un pair qui a commencé la lecture du contenu peut retransmettre à d'autres les segments qu'il a déjà reçus, et ainsi de suite, d'où une facilité de diffusion quel que soit le nombre de clients intéressés. Cette stratégie est décrite par exemple dans la demande internationale WO 2012/154287 et apporte satisfaction.

Toutefois, la plupart des lecteurs mettent en œuvre une logique dite de débit adaptatif (en anglais « Adaptive BitRate », ABR) et cela s'avère poser des problèmes lorsqu'en combinaison avec le P2P.

L'idée générale de l'ABR est de permettre la variation automatique de la qualité des segments récupérés en fonction des « capacités » d'un pair. Plus précisément, chaque segment est disponible à plusieurs niveaux de qualité correspondant à plusieurs bitrates, i.e. débits de données. On comprend en effet qu'un segment de meilleure qualité a une résolution meilleure, une compression moindre, un plus grand nombre d'images par secondes, etc., et est donc plus volumineux que le même segment dans une moindre qualité, et il est donc nécessaire de soutenir un débit de données plus élevé.

L'ABR vise à déterminer automatiquement pour chaque segment la qualité la meilleure pouvant être choisie, généralement au vu de deux critères qui sont la bande-passante constatée et/ou le taux de remplissage de la mémoire tampon.

Dans le premier cas, si l'algorithme juge que la bande passante estimée est suffisante pour supporter une qualité supérieure, alors il va donner l'instruction au client de basculer sur celle-ci (ou à l'inverse de baisser la qualité si la bande passante est trop faible). Dans le second cas, le principe est de diviser la mémoire tampon en différents intervalles, chaque intervalle correspondant à une qualité de plus en plus élevée à mesure que le remplissage de la mémoire tampon augmente (ou de plus en plus faible s'il baisse).

Dans les deux cas, si les algorithmes ABR n'ont pas d'incompatibilité fondamentale à être utilisés dans un contexte de streaming P2P, le problème est que les algorithmes d'ABR ont été conçus pour fonctionner dans un scénario de streaming simple, i.e. avec tous les segments récupérés sur requête depuis le serveur de contenus.

Or, en pratique le streaming P2P effectue du « pré-buffering », en téléchargeant des segments en P2P dans un cache P2P dédié avant que le lecteur n'en fasse réellement la requête. En effet, l'objectif du streaming P2P est

de solliciter le moins possible (et en dernier recours) le serveur de contenu d'origine : une requête directe d'un segment auprès de ce serveur n'est effectuée que s'il y a un risque qu'il n'y ait plus de segments dans le buffer vidéo et que la lecture s'interrompe (« re-buffering »), sinon on compte au maximum sur le réseau P2P.

On se retrouve ainsi du point de vue du lecteur avec des bandes-passantes apparentes extrêmement élevées, puisque des segments peuvent être chargés dans la mémoire tampon depuis le cache P2P une fraction de seconde après qu'ils ont été demandés. De surcroît le taux de remplissage du buffer vidéo est artificiellement élevé.

Cela provoque des décisions incontrôlées de l'ABR d'augmenter la qualité si la qualité courante n'est pas la qualité maximale, indépendamment de la capacité réelle du réseau, qualité qu'il ne pourra pas nécessairement supporter.

On se retrouve ainsi au mieux avec des oscillations instables de la qualité du flux, au pire avec des interruptions répétées de la lecture et des sollicitations nombreuses et inutiles du serveur de contenus.

Il serait souhaitable de pouvoir contrôler la logique d'ABR, mais malheureusement celle-ci est souvent propre au lecteur et peu accessible, et encore moins modifiable. Il serait ainsi souhaitable de disposer d'une manière de contrôler tout algorithme ABR dans un contexte de streaming P2P.

La présente invention vient améliorer la situation.

PRESENTATION DE L'INVENTION

La présente invention se rapporte ainsi à un procédé de lecture en continu sur un lecteur d'un équipement client d'un contenu diffusé au sein d'un réseau pair à pair d'équipements clients, ledit contenu étant constitué d'une séquence de segments disponibles dans une pluralité de niveaux de qualités et ledit lecteur étant adapté pour choisir le niveau de qualité des segments conformément à une logique de régulation adaptative du débit, ABR ; l'équipement client comprenant une première mémoire tampon adaptée pour stocker des segments dans un format adapté pour le transfert au sein du réseau pair-à-pair, le procédé étant caractérisé en ce qu'il comprend la mise en œuvre par des moyens de traitement de données de l'équipement d'étapes de :

- (a) Réception d'une requête d'un segment depuis le lecteur

- (b) Fourniture en réponse dudit segment depuis la première mémoire tampon à l'expiration d'un délai de réponse défini par rapport à ladite logique ABR du lecteur.

Cette méthode propose astucieusement un retour d'information à l'algorithme d'ABR par le biais de l'ajout d'un délai artificiel à la réponse.

Selon des caractéristiques avantageuses et non-limitatives :

- ladite logique ABR est définie par une première fonction permettant de calculer un niveau de qualité dans lequel ledit segment est requis en fonction d'au moins un paramètre représentatif d'un débit de réception de segments ;
- une deuxième fonction permet de calculer à l'étape (a) ledit délai de réponse en fonction dudit au moins un paramètre représentatif d'un débit de réception de segments ;
- la deuxième fonction est construite à partir de la première fonction ;
- la deuxième fonction est monotone par rapport audit au moins un paramètre représentatif d'un débit de réception de segments ;
- l'équipement comprend en outre une deuxième mémoire tampon adaptée pour stocker des segments dans un format adapté pour la lecture par le lecteur, ledit segment étant fourni à l'étape (b) dans la deuxième mémoire tampon ;
- ledit paramètre représentatif d'un débit de réception de segments est un niveau de remplissage de la deuxième mémoire tampon et/ou une bande-passante ;
- l'étape (a) comprend la prédiction d'un niveau de qualité dans lequel le prochain segment sera requis en fonction du délai de réponse calculé ;
- Le procédé comprend une étape (c) de vérification de ladite prédiction d'un niveau de qualité dans lequel le prochain segment sera requis ;
- Le procédé comprend l'apprentissage de la deuxième fonction sur la base des niveaux de qualité prédits et effectivement requis ;
- le segment requis est disponible de manière incomplète dans la première mémoire tampon, l'étape (a) comprenant la modification du délai de réponse calculé en fonction d'une durée estimée nécessaire pour finir de récupérer le segment, le segment étant fourni de manière complète à l'étape (b) au plus tôt à l'expiration du délai de réponse modifié.

Selon un deuxième aspect, l'invention propose un équipement pour la lecture en continu sur un lecteur d'un contenu diffusé au sein d'un réseau pair à pair d'équipements clients, ledit contenu étant constitué d'une séquence de segments disponibles dans une pluralité de niveaux de qualités et ledit lecteur étant adapté pour choisir le niveau de qualité des segments conformément à une logique de régulation adaptative du débit, ABR ; l'équipement client comprenant une première mémoire tampon adaptée pour stocker des segments dans un format adapté pour le transfert au sein du réseau pair-à-pair et des moyens de traitement de données, l'équipement caractérisé en ce que les moyens de traitement de données sont configurés pour, suite à la réception d'une requête d'un segment depuis le lecteur, fournir en réponse ledit segment depuis la première mémoire tampon à l'expiration d'un délai de réponse défini par rapport à ladite logique ABR du lecteur.

Selon un troisième et un quatrième aspect, l'invention propose un produit programme d'ordinateur comprenant des instructions de code pour l'exécution d'un procédé selon le premier aspect de diffusion d'un contenu en streaming dans un réseau pair à pair d'équipements clients connecté à un serveur de contenu ; et un moyen de stockage lisible par un équipement informatique sur lequel un produit programme d'ordinateur comprend des instructions de code pour l'exécution d'un procédé selon le premier aspect de diffusion d'un contenu en streaming dans un réseau pair à pair d'équipements clients connecté à un serveur de contenu.

PRESENTATION DES FIGURES

D'autres caractéristiques et avantages de la présente invention apparaîtront à la lecture de la description qui va suivre d'un mode de réalisation préférentiel. Cette description sera donnée en référence aux figures annexées dont :

- la figure 1 représente une architecture pour la mise en œuvre du procédé selon l'invention ;
- la figure 2 illustre un mode de réalisation préféré du procédé selon l'invention.

DESCRIPTION DETAILLEE

Architecture

En référence à la figure 1, l'invention propose un procédé de lecture en continu d'un contenu diffusé au sein d'un réseau 1 tel que représenté. Le réseau 1 est ici un réseau de télécommunications à grande échelle et en particulier Internet. Ce réseau 1 comprend un réseau pair-à-pair 10 d'équipements client 11, 12. Chaque équipement client 11, 12 est typiquement un équipement informatique personnel tel qu'un smartphone, un PC, une tablette, etc. connectée au réseau 1, disposant de moyens de traitement de données 110 tels qu'un processeur, d'une interface pour la lecture du contenu, et de moyens de stockage de données tels qu'une mémoire vive et/ou une mémoire de masse.

La lecture est mise en œuvre par un lecteur, c'est-à-dire une application exécutée par les moyens de traitement de données 110, qui peut être de nature variée, par exemple une application dédiée, un navigateur internet en particulier compatible HTML5, un module du système d'exploitation, etc.

On supposera dans la suite de la description que le lecteur est « en l'état », c'est-à-dire qu'il n'est pas modifié pour la mise en œuvre du présent procédé, ni même pour du streaming P2P. En particulier, le lecteur met en œuvre une logique de régulation adaptative du débit, ABR (Adaptative Bitrate), c'est-à-dire que ledit contenu à lire est constitué d'une séquence de segments disponibles dans une pluralité de qualités et que le lecteur est capable de décider de manière autonome quelle qualité requérir, conformément à cette logique ABR. Les diverses qualités correspondent à des débits différents, c'est-à-dire un volume de données par unité de temps (et ainsi par segment) variable. On comprend naturellement qu'un contenu de meilleure qualité nécessite un débit plus important.

On reviendra plus en détail plus loin sur la notion de logique ABR, on comprendra seulement que dans le cadre du présent procédé il n'est pas nécessaire que la logique ABR soit contrôlable ni même connue : le présent procédé est complètement universel et peut s'adapter à n'importe quel lecteur mettant en œuvre n'importe quelle logique d'ABR sur la base de n'importe quel critère. On supposera que la logique d'ABR est prédéfinie et que le logiciel client (voir plus loin) ne fait que la subir.

Par ailleurs, l'équipement 11 (et plus précisément ses moyens de stockage de données) disposent de deux mémoires tampon M1 et M2 (appelées également

« buffer »), typiquement deux zones d'une mémoire vive, chacune pouvant stocker (d'une façon différente comme l'on va voir) tout ou partie du contenu de façon temporaire (par temporaire, on entend que les segments sont supprimés de cette mémoire peu de temps après qu'ils aient été lus : il ne sont pas stockés à long terme comme c'est le cas pour un téléchargement direct). Par exemple, dans le cas d'une lecture via un navigateur, tous les segments sont typiquement supprimés (i.e. les mémoires tampon réinitialisées) au plus tard lorsque l'on ferme le navigateur ou l'onglet dans lequel la vidéo est jouée.

La première mémoire tampon M1 est appelée « cache pair-à-pair ». Elle stocke les segments sous un format dit « brut ». Par segments bruts, on entend sous un format adapté pour le transfert au sein du réseau pair-à-pair 10, mais inadapté pour la lecture sur l'équipement 11.

La deuxième mémoire tampon M2 est appelée « buffer vidéo ». Elle est contrôlée par le lecteur et stocke les segments sous un format dit « converti ». Par segments convertis, on entend convertis depuis des segments bruts sous un format adapté pour la lecture (et en particulier pour le lecteur) sur l'équipement 11, mais inadapté pour le transfert au sein du réseau pair-à-pair 10, voir plus loin.

Comme expliqué dans l'introduction, les équipements 11, 12 sont des « pairs » (appelés également « nœuds ») du réseau pair-à-pair 10.

Par « équipements client 11, 12 d'un réseau pair-à-pair 10 » on entend des équipements connectés dans le réseau 1 par un protocole réseau pair-à-pair. En d'autres termes, les moyens de traitement de données de chaque pair mettent en œuvre un programme particulier (logiciel client, appelé « Peer-agent », PA), qui peut être par exemple être intégré au lecteur (par exemple une extension d'un navigateur web), faire l'objet d'une application dédiée, voire être incorporé dans tout autre logiciel embarqué (par exemple un player d'un boîtier d'accès à Internet ou d'un boîtier multimédia, i.e. une « Set-top box »), pour l'utilisation du pair-à-pair. Le présent procédé est principalement implémenté via ce logiciel client. Dans la suite de la présente description, on fera l'hypothèse d'un logiciel client en communication avec le lecteur de sorte à l'approvisionner en segments, mais fonctionnant de manière indépendante. Plus précisément, on comprend que le rôle du lecteur est la lecture en elle-même, i.e. le rendu des segments, alors que le rôle du logiciel client est simplement l'obtention des segments pour le lecteur, le

logiciel client subissant le fonctionnement du lecteur et en particulier sa logique ABR.

Comme expliqué, un réseau pair-à-pair, ou « peer-to-peer » ou encore P2P, est un sous-réseau décentralisé au sein du réseau 1, dans lequel des données peuvent être transférées directement entre deux équipements client 11, 12 du réseau 10, sans transiter par un serveur central. Il permet ainsi à tous les équipements client 11, 12 de jouer à la fois le rôle de client et serveur. Les pairs 11, 12 sont ainsi définis comme des « seeders » (en français les « semeurs », c'est-à-dire des fournisseurs de données) et/ou des « leechers » (en français les « sangsues », c'est-à-dire des receveurs de données).

Ledit contenu, qui est en particulier un contenu audio ou vidéo, c'est-à-dire un média d'une certaine durée, est constitué d'une séquence de segments (dite « liste de lecture », en anglais « playlist ») stockés sur des moyens de stockage de données d'un serveur 2 connecté au réseau pair-à-pair 10, et comme expliqués disponibles en plusieurs niveaux de qualité. Les segments ont une longueur prédéterminée, typiquement une ou deux secondes du contenu, mais elle peut aller d'une fraction de seconde à une dizaine de secondes. Tous les segments d'un contenu donné ont généralement la même longueur, qu'on notera ici p.

Le serveur 2 est un serveur de contenu (dit CDN), avantageusement présent dans le réseau 1 et connecté au réseau pair-à-pairs 10. En d'autres termes, il s'agit d'un (ou plusieurs) serveurs du réseau internet 1 mettant à disposition les segments de divers contenus conformément à un protocole de streaming donné. On citera par exemple l'exemple de HLS (« HTTP Live Streaming »), dans lequel les segments sont des fichiers « ts », listés dans un fichier playlist « m3u8 ». HLS implique le format MPEG2 pour le contenu. On citera également les protocoles de streaming DASH, Smooth streaming, ou HDS. Les segments bruts peuvent être échangés entre pairs via un protocole de type WebRTC.

Le serveur 2 est la source primaire des segments, dans la mesure où initialement aucun pair ne dispose du contenu (avant un premier transfert du serveur 2 vers ce pair 11, 12). Les contenus sont soit dès l'origine stockés dans leur intégralité sur le serveur 2 (cas de la VOD précédemment évoqué), soit générés en temps réel (cas du live streaming, ou streaming en direct), et dans ce dernier cas la liste de segments qui les constitue évolue dynamiquement.

Le « Live Streaming » (c'est-à-dire le « streaming en direct ») propose de diffuser en temps réel des contenus associés à des événements « en live », par exemple des concerts, des conférences, des rencontres sportives, des parties de jeux vidéo, etc., qui sont en train de se dérouler simultanément. Par rapport au streaming d'un contenu déjà intégralement existant tel qu'un film, un contenu diffusé en Live streaming est en effet généré au fur et à mesure que l'événement associé se déroule. Techniquement, comme dans le cas d'un direct à la télévision, un tel contenu ne peut être diffusé qu'avec un léger retard, que l'utilisateur souhaite le plus faible possible. Ce retard est typiquement de l'ordre d'une minute, mais peut descendre jusqu'à une vingtaine de secondes. Ceci fait qu'une liste de lecture de seulement une poignée de segments (tout au plus quelques dizaines) est disponible à chaque instant, les segments de cette liste étant renouvelés dynamiquement selon un roulement : au fur et à mesure que l'événement se déroule des nouveaux segments sont créés, « vieillissent », sont reçus et lus par les clients (à l'issue du retard prévu), et enfin sortent de la liste.

Dans ce dernier cas (live streaming), le contenu doit plutôt être vu comme un flux en continu. La séquence de segments est ainsi dynamique, c'est-à-dire qu'elle est mise à jour régulièrement. A chaque fois qu'un nouveau segment est généré il est ajouté en fin de la séquence, et le premier segment de la séquence (le plus ancien) est supprimé. Tous les autres se décalent selon un mécanisme de roulement que l'on peut apparenter à une liste FIFO. Le premier segment de la liste (le plus ancien) peut être celui au niveau du point de lecture, en d'autres termes le segment « en direct » (et donc les segments sont supprimés de la liste de lecture dès qu'ils sont lus), ou un segment « passé » si le serveur de contenu accepte qu'on lise le contenu avec du retard (certaines plateformes proposent du live streaming avec jusqu'à 2h de retard), c'est ce que l'on appelle le DVR (« enregistreur vidéo numérique »).

Le présent procédé peut être mis en œuvre dans n'importe quel contexte.

Est également avantageusement connecté au réseau pair-à-pair 10 un serveur de gestion de pairs 3 appelé « tracker ». Le tracker 3 présente des moyens de traitement de données et des moyens de stockage. Il coordonne les échanges entre pairs 11, 12 (en contrôlant le logiciel client mis en œuvre par chacun des équipements client 11, 12), mais il n'est pas directement impliqué dans le transfert de données et ne possède pas de copie du fichier.

ABR

Comme évoqué il y a deux types principaux de logiques ABR, basés respectivement sur la bande passante (i.e. le débit de données constaté) et sur le niveau de remplissage (soit en valeur, i.e. en secondes ou en nombre de segments, soit en taux) de la deuxième mémoire tampon M2 (éventuellement en combinaison). En d'autres termes, le lecteur monitoré la bande-passante et/ou le volume/taux de remplissage de la deuxième mémoire tampon M2, et prend en conséquence des décisions de modification ou non du niveau de qualité des segments exigé.

Généralement, dans un cas comme l'autre on peut définir la logique ABR au moyen d'une première fonction permettant de calculer le niveau de qualité à choisir (le bitrate) en fonction d'au moins un paramètre représentatif d'un débit de réception de segments, voir par exemple la courbe telle que représentée par la **figure 2** (partie gauche, qui représente le côté lecteur, la partie droite illustrant le côté logiciel client).

On comprend que ledit paramètre représentatif d'un débit de réception de segments est un paramètre surveillé, qui peut être n'importe quel paramètre illustrant la capacité de l'équipement 11 et/ou du réseau 10 à recevoir « suffisamment vite » les segments. Comme expliqué précédemment, les logiques ABR connues utilisent généralement comme paramètre un niveau de remplissage de la deuxième mémoire tampon M2 et/ou une bande passante.

Quel que soit le type de logique ABR, ladite première fonction est monotone, en particulier croissante, entre une valeur de qualité minimale et une valeur de qualité maximale. Un modèle préféré est celui d'une fonction continue par morceaux, à trois morceaux (connectés en deux points singuliers), dont un premier et un troisième morceaux constants.

Ainsi, dans l'exemple de la figure 2 (partie gauche), l'exemple d'une logique ABR basée sur le niveau de remplissage de la deuxième mémoire tampon M2 (exprimé en secondes) est représenté, et on voit que :

- pour un niveau de remplissage inférieur ou égal à un seuil minimal $B_{\min}$, le bitrate est fixé égal au bitrate minimum $R_{\min}$ (en d'autres termes le buffer est dangereusement vide, ce qui est représentatif de difficultés à

obtenir des segments, et alors la qualité est minimale pour favoriser le re-remplissage) ;

- pour un niveau de remplissage supérieur ou égal à un seuil maximal $B_{\max}$, le bitrate est fixé égal au bitrate maximum $R_{\max}$ (le buffer est plein donc il n'y a pas de risque à se permettre la qualité maximale) ;
- entre les deux, on a une courbe croissante, préférentiellement linéaire – dans l'exemple de la figure 2 elle est d'équation $R=R_{\min}+(B-B_{\min})*(R_{\max}-R_{\min})/(B_{\max}-B_{\min})$ – ce qui traduit que plus le buffer est plein plus la qualité peut être élevée.

Une courbe similaire est obtenue pour une logique ABR basée sur la bande-passante

Comme l'on verra plus loin, le présent procédé propose de contrôler l'ABR non pas en altérant sa logique, mais en la leurrant au moyen d'un délai de réponse rajouté avant de satisfaire une requête du segment. En effet, simplement retarder la fourniture d'un segment permet de limiter artificiellement la bande-passante et le niveau de remplissage de la deuxième mémoire tampon M2, et ainsi d'agir sur la qualité attendue des segments.

On définit pour cela une deuxième fonction permettant ledit délai de réponse en fonction dudit au moins un paramètre représentatif d'un débit de réception de segments dans les exemples représentés, c'est donc à nouveau le niveau de remplissage de la deuxième mémoire tampon M2).

La deuxième fonction est préférentiellement construite (voire apprise, voir plus loin) à partir de la première fonction, c'est-à-dire qu'on peut mapper la logique ABR au moyen de la deuxième fonction.

Ainsi, selon un mode de réalisation préféré la deuxième fonction est monotone par rapport audit au moins un paramètre représentatif d'un débit de réception de segments, et généralement (notamment dans le cas d'une logique ABR basée sur le niveau de remplissage de la deuxième mémoire tampon M2 qui sera développé ci-après) la première et la deuxième fonction ont la même monotonie par rapport audit au moins un paramètre représentatif d'un débit de réception de segments (en particulier croissante). En effet la deuxième fonction s'avère le plus souvent similaire à la première fonction de la logique ABR (à nouveau une fonction continue par morceaux à trois morceaux dont deux

segments extrémaux), voir par exemple la courbe correspondante qui est représentée sur la partie droite de la figure 2.

Le mapping peut par exemple consister à prendre deux points de la première fonction, en particulier les points singuliers connectant les morceaux de coordonnées $\{B_{\min}, R_{\min}=f_1(B_{\min})\}$ et $\{B_{\max}, R_{\max}=f_1(B_{\max})\}$, et construire la deuxième fonction sur la base des deux points correspondants $\{B_{\min}, \rho_{\min}=f_2(B_{\min})\}$ et $\{B_{\max}, \rho_{\max}=f_2(B_{\max})\}$. De manière préférée on construit la deuxième fonction de même monotonie allure que la première fonction, en particulier une fonction continue par morceaux à trois morceaux connectés via lesdits deux points.

Dans l'exemple de la figure 2 partie droite, la deuxième fonction est ainsi construite de la sorte :

- pour un niveau de remplissage inférieur ou égal au seuil minimal $B_{\min}$, le délai de réponse est fixé à une valeur minimale $\rho_{\min}$, préférentiellement nulle, i.e. aucun délai n'est ajouté (en d'autres termes le buffer est dangereusement vide, et on ne prend pas de risque en fournissant directement les segments reçus) ;
- pour un niveau de remplissage supérieur ou égal à un seuil maximal $B_{\max}$, le délai de réponse est fixé égal à une valeur maximale $\rho_{\max}$, préférentiellement la longueur p d'un segment (ce permet de ne pas augmenter encore davantage le taux de remplissage du buffer vidéo, le cache P2P se garde de la marge en ne fournissant un nouveau segment que lorsqu'un segment précédent aura été lu en entier) ;
- entre les deux, on a une courbe croissante, préférentiellement linéaire – dans l'exemple de la figure 2 elle est d'équation $\rho = \rho_{\min} + (B - B_{\min}) * (\rho_{\max} - \rho_{\min}) / (B_{\max} - B_{\min})$ – ce qui traduit que plus le buffer est plein plus le délai de réponse peut être élevé de sorte à prévenir une hausse supplémentaire de ce taux de remplissage.

A noter qu'il est également possible que la première et la deuxième fonction aient une monotonie inverse par rapport audit au moins un paramètre représentatif d'un débit de réception de segments (c'est-à-dire que la deuxième fonction est croissante lorsque la première fonction est décroissante, et vice-versa), c'est notamment le cas si la logique ABR est basée sur la bande passante (alors la première fonction est croissante et la deuxième fonction est décroissante).

Le présent procédé ne sera pas limité à une stratégie donnée d'application d'un délai de réponse, et l'homme du métier saura adapter l'idée à la logique ABR, au type de paramètre surveillé par le lecteur, au comportement du réseau P2P, etc. De façon générale, on comprendra qu'il suffit de construire une deuxième fonction définissant le délai de réponse à partir de la première fonction définissant la logique ABR.

Lecture

Dans la suite de la présente description, on se focalise sur un équipement client 11 qui est le cas échéant en train de récupérer le contenu depuis d'autres équipements 12 et/ou le serveur 2, c'est-à-dire que la première mémoire tampon M1 stocke d'ores et déjà au moins un segment brut, dans au moins un niveau de qualité, si possible une sous-séquence de la séquence constituant le contenu.

Le procédé commence alors par la mise en œuvre par les moyens de traitement 110 de l'équipement d'une étape (a) de réception d'une requête d'un segment, en l'espèce du prochain segment à mettre en deuxième mémoire tampon M2 (pas nécessairement le prochain segment à lire, il y a normalement des segments d'avance bufferisés). Ladite requête est reçue depuis le lecteur, et définit préférentiellement le niveau de qualité dans lequel le segment est requis, i.e. le bitrate (par application de la logique ABR).

On suppose que ledit segment est à ce stade au moins partiellement disponible (i.e. au moins un fragment) dans la première mémoire tampon M1, dans la qualité requise par le lecteur. Si ce segment/fragment de segment était dans une autre qualité, il faudrait le récupérer à nouveau, généralement directement depuis le serveur de contenu 2 car on manque de temps.

Comme expliqué, l'étape (a) comprend alors avantageusement le calcul d'un délai de réponse défini par rapport à ladite logique ABR du lecteur, en particulier en fonction d'un paramètre surveillé par la logique ABR (ledit au moins un paramètre représentatif d'un débit de réception de segments), préférentiellement le niveau de remplissage de la deuxième mémoire tampon M2 et/ou la bande-passante. L'étape (a) comprend le cas échéant la « mesure » dudit paramètre représentatif d'un débit de réception de segments.

Comme expliqué, la deuxième fonction définissant le délai de réponse est préférentiellement croissante avec ledit paramètre représentatif d'un débit de

réception de segments entre une valeur minimale de délai de réponse et une valeur maximale.

Dans le cas où c'est seulement un fragment du prochain segment qui a été récupéré (on dit que le segment est disponible de manière incomplète), de manière préférée le délai de réponse calculé est modifié selon la longueur du fragment de sorte à refléter le fait que seul un fragment du délai de réponse doit réellement être appliqué. En effet, on ne peut fournir à la deuxième mémoire tampon M2 que des segments complets et non des fragments, et l'idée est de fournir le segment de manière complète à l'issue d'un délai de réponse raccourci reflétant le fait qu'il y aura déjà implicitement un délai d'attente correspondant au temps pour compléter (finir de récupérer) ce segment dans la première mémoire tampon M1. Ainsi, l'étape (a) comprend la modification du délai de réponse calculé en fonction d'une durée estimée nécessaire pour finir de récupérer le segment.

Par exemple, on pourra appliquer la formule $p' = p - tdw$, où p' est le délai de réponse modifié et tdw est la durée estimée nécessaire pour finir de récupérer le segment. Ainsi, l'attente du temps tdw plus l'application de p' avant de fournir le segment complet est équivalent à l'application de p , de sorte que le délai global reste le même.

Dans une étape (b), ledit segment requis est fourni en réponse à la requête, depuis la première mémoire tampon M1, à l'expiration dudit délai de réponse. Par « fourni à l'expiration dudit délai de réponse », on entend de telle sorte que le lecteur n'en dispose pas avant l'expiration du délai de réponse (au mieux au moment de l'expiration, voire seulement après dans certains cas, voir plus loin). Le plus souvent, le segment est transmis d'un coup au moment de l'expiration du délai de réponse, mais on comprendra qu'il est tout à fait possible de le « streamer » au sein de l'équipement 1, i.e. de le transmettre depuis la première mémoire tampon M1 progressivement (petit bout par petit bout) de sorte que le dernier bout est transmis (au plus tôt) au moment de l'expiration du délai de réponse (le délai de réponse est alors un « délai de transmission du dernier bit du segment »). En effet, bien que seuls des segments complets sont lisibles, certains lecteurs peuvent accepter des sous-segments du segment. A noter qu'une telle transmission progressive ne change rien puisque tant que le segment n'est pas intégralement reçu il n'est pas disponible par le lecteur et donc pas considéré comme fourni, mais permet de faciliter des mesures de bande-passante.

Dans le cas où seul un fragment du segment était disponible dans la première mémoire tampon M1 et où le délai de réponse a été modifié en fonction d'une durée estimée nécessaire pour finir de récupérer le segment, normalement le segment est également fourni à l'étape (b) à l'expiration du délai de réponse modifié. Comme expliqué, bien que la fourniture puisse être morcelée, il ne faut pas confondre des sous-segments d'un segment complet (qui correspondent à des morceaux consécutifs de segment obtenus à partir d'un segment complètement téléchargé) et un segment incomplet (dans lequel seules certaines parties de données, correspondant le plus souvent à des morceaux disparates, ont été téléchargées). Seul un segment disponible de manière complète dans la première mémoire tampon M1 peut être fourni (le cas échéant progressivement) en réponse à la requête (et pas un fragment), de sorte que si le téléchargement a pris plus de temps de prévu, il est possible que le segment ne soit disponible de manière complète qu'après l'expiration du délai de réponse modifié. Ainsi, le segment complet est fourni au plus tôt à l'expiration du délai de réponse modifié (i.e. pas avant), mais possiblement après. En pratique, le segment complet est fourni lorsque les deux conditions suivantes sont vérifiées : le segment est disponible de manière complète (son téléchargement est terminé), et le délai de réponse modifié a expiré.

Dans tous les cas, de manière préférée, le segment est fourni à la deuxième mémoire tampon M2, et à ce titre l'étape (b) peut comprendre la conversion dans un format adapté pour la lecture sur l'équipement 11 dudit segment de la première mémoire tampon M1. Cette étape consiste à transformer le segment brut en un segment converti, qui pourra être lu par le lecteur de l'équipement 11 contrairement au premier.

Par exemple, si le lecteur est le lecteur intégré d'un navigateur compatible HTML5, la conversion consiste en l'injection des données vidéo du segment grâce à l'API Media Source Extension du navigateur.

Naturellement, l'étape (b) comprend avantageusement simultanément la lecture d'un segment précédent stocké dans la deuxième mémoire tampon M2, de sorte que les segments « tournent ». Le segment récupéré dans l'étape (b) sera bientôt lu à son tour.

On peut à présent répéter les étapes (a) et (b) tant que la lecture dure : l'application du délai stabilise l'ABR, et au pire, en cas de changement quand même du niveau de qualité requis par le lecteur du fait d'une nouvelle décision

ABR, les segments seront à présent chargés depuis le réseau P2P conformément au nouveau niveau de qualité requis, de sorte que l'utilisateur n'aura aucune gêne.

Prédiction

De manière particulièrement préférée, l'étape (a) comprend en outre la prédiction d'une éventuelle prochaine décision ABR, i.e. la prédiction du niveau de qualité dans lequel le prochain segment sera requis (i.e. à la prochaine occurrence de l'étape (a)).

En effet, il est possible par application de la première fonction, de déterminer avant d'appliquer le délai de réponse calculé, si le niveau de qualité requis par l'ABR va changer.

On va supposer qu'un segment complet, de longueur p , a été fourni de sorte à faciliter les calculs. Ainsi, à l'issue de l'étape (b), le niveau de remplissage de la deuxième mémoire M2 aura augmenté de p , c'est-à-dire la longueur du segment. Par contre, le délai de réponse p s'est écoulé, et une durée identique du contenu de la deuxième mémoire M2 a été lue et supprimé, de sorte que le nouveau niveau de remplissage B_n peut être exprimé par rapport au précédent B_{n-1} (au moment de la dernière fourniture d'un segment) par la formule $B_n = B_{n-1} + p - p_n$ (on note qu'un calcul similaire peut être fait si le paramètre représentatif du débit de réception de segments est la bande-passante). Il suffit donc de calculer $f_1(B_n)$ pour prédire la prochaine décision d'ABR, et le cas échéant anticiper un changement de niveau de qualité et requérir les futurs segments dans le bon niveau de qualité (on parle de « pre-fetch »). Ainsi, la probabilité que l'utilisateur soit gêné est encore plus faible.

A noter que le procédé peut comprendre une étape (c), à l'issue de l'étape (b), de vérification de la prédiction (le cas échéant confondue avec une prochaine occurrence de l'étape (a)).

Plus précisément, la logique ABR est relancée pour calculer un « nouveau » niveau de qualité requis, i.e. le niveau de qualité requis pour le prochain segment, et une requête de segment correspondant va à ce titre être envoyée au logiciel client. La vérification consiste simplement à comparer le niveau de qualité prédit et le niveau de qualité en pratique requis, pour le prochain segment.

Apprentissage

Selon un premier mode de réalisation, on suppose connue la première fonction, et la deuxième fonction peut être préconstruite à partir de cette première fonction.

Cependant dans d'autres cas, la première fonction peut être mal connue voire pas connue du tout si le lecteur est complètement fermé.

Alors, il est possible d'apprendre progressivement la première fonction et/ou la deuxième fonction, en se servant notamment de la prédiction de décision d'ABR comme moyen d'apprentissage : à chaque segment, on vient comparer à l'étape (c) la prédiction et la décision réellement prise, i.e. le niveau de qualité prédit et celui effectivement requis, ce qui constitue une boucle d'apprentissage.

On peut estimer la première ou la deuxième fonction en partant notamment d'une fonction par défaut, et en faisant évoluer ses paramètres au cours du temps. Par exemple, la première fonction peut être paramétrée via les valeurs des quatre paramètres $B_{\min}$, $R_{\min}$, $B_{\max}$ et $R_{\max}$ en supposant une fonction par morceaux à trois segments, et similairement la deuxième fonction peut être paramétrée via les valeurs des quatre paramètres $B_{\min}$, $\rho_{\min}$, $B_{\max}$ et $\rho_{\max}$ en supposant une fonction par morceaux à trois segments (on a donc au total six variables).

Ainsi, sur les premiers segments du contenu le logiciel client va probablement calculer le délai de réponse manière imprécise, ce qui entraînera des décisions ABR imprévues et changement de niveau de qualité non maîtrisées, mais rapidement l'algorithme va s'améliorer et la logique ABR sera complètement maîtrisée.

Des algorithmes d'apprentissage peuvent notamment être utilisés, par exemple des réseaux de neurones.

A noter que les données sur la base desquelles l'apprentissage est fait peuvent être conservées entre sessions pour éliminer ou limiter les problèmes liés à la phase d'apprentissage lors des sessions subséquentes.

Certains facteurs, comme par exemple un changement sur le numéro de version du lecteur, peuvent être utilisés pour invalider les paramètres sauvegardés et repartir sur des paramètres vierges pour une nouvelle phase d'apprentissage.

Alternativement/en supplément, les fonctions apprises peuvent être transmis à d'autres pairs 12, par exemple via des messages P2P, ou au serveur de gestion de pairs 3.

Équipement et produit programme d'ordinateur

Selon un deuxième aspect, l'invention concerne l'équipement client 11 pour la mise en œuvre du présent procédé de lecture d'un contenu, sur un lecteur adapté pour choisir le niveau de qualité des segments conformément à une logique de régulation adaptative du débit, ABR.

Cet équipement 11 comprend comme expliqué :

- Une première mémoire tampon M1 stockant de façon temporaire au moins un segment d'un contenu constitué d'une séquence de segments, chaque segment brut étant dans un format adapté pour le transfert au sein du réseau pair-à-pair 10 ;
- Une deuxième mémoire tampon M2 stockant de façon temporaire au moins un segment converti dudit contenu, dans un format adapté pour la lecture par le lecteur ; et
- des moyens de traitement de données 110.

Les moyens de traitement de données 110, typiquement un processeur, sont configuré pour implémenter, suite à la réception d'une requête d'un segment, la fourniture (au lecteur, typiquement par injection dans la deuxième mémoire tampon M2) dudit segment dans la première mémoire tampon M1 à l'expiration d'un délai de réponse défini par rapport à ladite logique ABR du lecteur.

Selon d'autres aspects aspect, l'invention concerne un produit programme d'ordinateur comprenant des instructions de code pour l'exécution (sur des moyens de traitement de données, en particulier ceux de l'équipement client 11) d'un procédé selon le premier aspect de l'invention de lecture en continu sur un équipement client 11 d'un contenu diffusé au sein d'un réseau pair à pair 10 d'équipements clients 11, 12, ainsi que des moyens de stockage lisibles par un équipement informatique (par exemple une mémoire de cet équipement client 11) sur lequel on trouve ce produit programme d'ordinateur.