HAL open science

Optimisation de la retransmission des paquets perdus en streaming P2P

Houssein Wehbe, Gérard Babonneau, Bernard Cousin

To cite this version:

Houssein Wehbe, Gérard Babonneau, Bernard Cousin. Optimisation de la retransmission des paquets perdus en streaming P2P. JDIR, Mar 2010, Sophia Antipolis, France. inria-00470547

HAL Id: inria-00470547

https://inria.hal.science/inria-00470547v1

Submitted on 6 Apr 2010

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.

Optimisation de la retransmission des paquets perdus en streaming P2P

Houssein Wehbe, et Gérard Babonneau
Orange Labs
4, rue du Clos Courtel
Cesson Sévigné, France, 35512
Email: {houssein.wehbe, gerard.babonneau}@orange-ftgroup.com

Bernard Cousin
IRISA / Université de Rennes 1
Campus de Beaulieu
Rennes, France, 35042
Email: Bernard.cousin@irisa.f

Résumé— **Les réseaux pair-à-pair (P2P) jouent aujourd'hui un rôle primordial dans la livraison de la vidéo à grande échelle. La perte des paquets sur les réseaux IP, peut affecter défavorablement la qualité de la vidéo reçue. Différents mécanismes de réparation de perte sont utilisés aujourd'hui. Ces mécanismes ont été proposés initialement pour des applications n'ayant pas de contraintes de délai de réparation. Ils ne peuvent pas, donc, garantir la qualité de la vidéo en cas de perte. Nous proposons dans ce papier un nouveau mécanisme de réparation de perte. Il permet d'optimiser la qualité de la vidéo transmise sur les réseaux P2P. Son principe consiste à demander les paquets perdus à un pair sélectionné aléatoirement parmi les pairs du réseau ayant ces paquets. Ce pair ne sera pas systématiquement leur fournisseur initial. Ce mécanisme vise à augmenter la probabilité de choisir un pair disponible pour faire la retransmission. Ceci peut augmenter la probabilité de recevoir les paquets retransmis avant leur temps de visualisation. Ce qui améliore la qualité de la vidéo reçue. Nos études par simulations ont montré l'efficacité de ce mécanisme par rapport aux mécanismes actuels de retransmission.**

Mots-clès: réseau pair-à-pair, transmission vidéo, perte des paquets, retransmission de paquets.

I. INTRODUCTION

Ces dernières années ont vu une avancée fulgurante dans la transmission des données multimédia sur les réseaux IP. La part de la diffusion de flux vidéo de très bonne qualité devient de plus en plus importante, ce qui engendre des contraintes sur les réseaux en termes de débit, de temps de latence et de gigue.

L'émergence des systèmes de type pair-à-pair (P2P) dans le contexte de la distribution de la vidéo (aussi appelés systèmes de streaming en P2P) permet une amélioration importante des performances par rapport aux architectures basées sur des serveurs centralisés, principalement en termes de scalabilité (aussi appelée passage à l'échelle). Le principe fondamental des architectures P2P est l'équivalence (à priori) de rôle entre toutes les entités qui composent le système, qui sont appelées pairs ou nœuds. En plus de tous les problèmes hérités du protocole IP (bande passante non garantie, présence de perte dans le réseau, délai de transmission et gigue non bornés), les problèmes de QoS dans les réseaux P2P sont liés aux comportements dynamiques des nœuds participants au réseau. En effet, dans un réseau P2P, un nœud décide à lui seul quand rejoindre ou quitter le réseau. Il peut donc décider de ne plus partager ses ressources, ou aussi de limiter la bande passante remontante de son lien d'accès utilisée pour la vidéo selon une priorité attribuée à des services qu'il gère. Même si ces choix améliorent la qualité des services en limitant les congestions, cela ne supprime pas totalement les pertes des données. Ces pertes affectent défavorablement la qualité de la vidéo reçue.

Dans un grand nombre des systèmes de streaming en P2P, la vidéo est découpée en fragments (appelés « chunks »). Leur taille est adaptée en fonction du système P2P. Pour obtenir une vidéo de bonne qualité, le récepteur doit recevoir tous ses fragments avant leur temps d'affichage sur l'écran. Généralement, un fragment est transmis sur internet dans plusieurs paquets IP. En absence d'une technique d'autocorrection (dont la mise en œuvre est toujours coûteuse), la perte d'au moins un paquet rend ce fragment non exploitable. Pour assurer la qualité de la vidéo, il faut faire appel à une technique de réparation des paquets perdus. Cette technique doit garantir la réparation avant le temps d'affichage du fragment affecté par la perte.

Les systèmes actuels de streaming en P2P [1, 2, 3, 4, 5] ne proposent pas de techniques de réparation de perte spécifiques. Les techniques utilisées ne prennent pas en compte les contraintes temporelles de la vidéo (elles ont été proposées initialement pour le transfert des fichiers). Elles consistent à demander ces paquets au pair qui les a émis initialement. Cependant, si ce pair n'était pas disponible à la réception de la requête de retransmission, la probabilité de ne pas recevoir les paquets retransmis à temps sera grande. Ce qui peut affecter la qualité de la vidéo. Pour cela, nous supposons que le fournisseur initial des paquets n'est pas toujours le pair le plus apte pour les retransmettre en cas de pertes. Nous proposons de demander la retransmission de ces paquets à un pair sélectionné aléatoirement parmi les pairs qui en disposent, et qui ne sera pas systématiquement le fournisseur initial. Ce mécanisme vise à sélectionner un fournisseur de retransmission disponible. Le choix d'un tel fournisseur peut augmenter la probabilité de recevoir les paquets retransmis à temps pour améliorer la qualité de la vidéo reçue. Notre mécanisme de retransmission peut fonctionner en complément de n'importe quel système de streaming en P2P. En effet, il ne nécessite pas

d'ajouter de nouveaux messages de signalisations, ni de modifier les algorithmes de distribution de la vidéo définis dans ces systèmes (comme ceux de localisation des fragments). Dans la mesure où ce procédé est indépendant des pairs émetteurs, il peut aussi s'appliquer à une diffusion multicast, en répartissant les requêtes du groupe sur des pairs au lieu de tout envoyer à la source.

Ce papier est organisé de la manière suivante: la section 2 présente un état de l'art des mécanismes de réparation de perte utilisés dans les systèmes actuels de streaming en P2P. La section 3 présente la solution proposée avant que la section 4 introduise les tests de performance réalisés. Finalement la section 5 conclut et indique les travaux à venir.

II. ETAT DE L'ART

Les schémas de réparation des pertes utilisés dans les systèmes de streaming en P2P s'appuient sur la retransmission ou la correction d'erreur directe (FEC "Forward Error Correction").

La retransmission des paquets perdus est un moyen simple. Elle s'applique nativement si le système P2P utilise le protocole TCP ("Transmission Control Protocol") pour le transport des données. En revanche, si le protocole utilisé est UDP ("User Datagram Protocol"), une technique de demande de retransmission doit être ajoutée au niveau applicatif.

Si le protocole utilisé est TCP, le récepteur doit envoyer à l'émetteur un acquittement de chaque paquet reçu. L'émetteur utilise l'absence d'acquittements pour détecter les paquets perdus et les retransmettre. Mais étant donné que TCP considère les pertes de paquets comme des signaux de congestion de réseau, il réduit systématiquement son débit d'émission à la détection d'une perte pour éviter de saturer le réseau. La vidéo peut être retardée au point de provoquer des gels d'images. De plus, dans les réseaux asymétriques (comme le cas des réseaux P2P utilisant, par exemple, les réseaux ADSL), si le canal remontant du récepteur est encombré, les acquittements peuvent être perdus ou retardés. Dans ce cas aussi, l'émetteur TCP réduit son débit d'émission. Pour ces raisons, TCP n'est pas souvent approprié pour la transmission de la vidéo en temps réel. Le protocole UDP n'a pas ces inconvénients car il maintien son débit, mais il ne définit aucun mécanisme pour récupérer les paquets perdus.

Pour appliquer la retransmission avec le protocole UDP, il faut définir une technique permettant au récepteur de détecter la perte. On peut par exemple utiliser le protocole applicatif RTP [13]. Ce protocole permet à l'émetteur de numéroter les paquets avant de les envoyer. Le récepteur peut donc envoyer une demande de retransmission à la détection d'une perte. Cependant, avec ce simple mécanisme de demande de retransmission, les récepteurs n'ont aucune garantie sur le temps de réparation.

Une autre technique de réparation de perte est utilisée avec UDP. C'est la correction d'erreur directe (FEC). La FEC est un moyen par lequel des données de redondance sont ajoutées au flux transmis, de telle sorte que les paquets perdus puissent être réparées par le récepteur de ce flux sans autre recours à la source. La source doit connaître le taux et les profils de perte pour ajouter en permanence des données de réparation suffisantes pour protéger le flux. Cependant, dans les réseaux P2P où les liens sont hétérogènes et le comportement des nœuds est imprévisible, la source ne peut pas connaître le taux de perte dans tout le système. En général, la quantité d'informations de FEC est un compromis entre la redondance d'information et le niveau des pertes qu'il est possible de corriger.

La plupart des systèmes actuels de streaming en P2P utilisent le protocole TCP pour le transport des données (voir le Tableau 1). Les systèmes utilisant le protocole UDP ne propose pas de mécanismes de réparation de perte spécifiques. Une amélioration de la qualité de la vidéo transmise en P2P a été proposé dans [9]. Elle consiste à protéger, par la FEC et/ou par retransmission, les paquets les plus importants de d'une vidéo de type JPEG. Selon les auteurs, ces sont les paquets appartenant aux images de type I et P d'un groupe d'images (GOP "Group Of Pictures"). En effet, les images de ces deux types sont plus importantes que les images de troisième type car ces troisièmes dépendent des deux premières par leur décodage. De même les images P dépendent des images I. Les résultats présentés dans [9] montrent que cette technique de protection améliore la qualité du flux, mais les analyses ont été effectuées sur un système P2P où le récepteur ne reçoit la vidéo que d'un seul pair. Ces résultats ne sont donc valables que dans ce cas particulier.

TABLEAU I. LES PROTOCOLES DE TRANSPORT ET LES MECANISMES DE REPARATION DE PERTE UTILISES PAR QUELQUES SYSTEMES DE STREAMING EN P2P

Système de streaming en P2P	***Protocole de transport***	***Réparation des pertes***
PPLive [2]	TCP	Retransmission
PULSE [1]	TCP	Retransmission
P2VoD [5]	TCP	Retransmission
PROMISE [6]	UDP	FEC

Nous proposons dans la section suivante un nouveau mécanisme de retransmission des paquets perdus. Ce mécanisme peut être appliqué dans n'importe lequel des systèmes de streaming en P2P utilisant le protocole UDP.

III. LE MECANISME DE RETRANSMISSION PROPOSE

La présence de plusieurs nœuds dans le système P2P disposant du même fragment de la vidéo, pose le problème de la sélection du meilleur pair pouvant assurer la meilleure qualité pour le service de transmission du fragment au récepteur, en particulier, pour minimiser le délai de réception. Il existe plusieurs métriques objectives ou subjectives qui peuvent servir à sélectionner les meilleurs nœuds, telles que, la mesure de la bande passante disponible entre les nœuds sources et le nœud récepteur, la bande passante disponible du nœud source, le délai de transmission entre la source et le récepteur (« one way delay »), le délai aller-retour (RTT), le nombre de

sauts, etc. Chaque système de streaming en P2P définit ses propres métriques de sélection. Généralement, des mécanismes de monitoring peuvent nous aider à l'estimation de ces métriques.

Après avoir sélectionné le pair fournisseur d'un fragment donné, une requête lui sera adressée. Il répond en envoyant le fragment dans des paquets IP. Généralement, un fragment est beaucoup plus gros qu'un paquet IP[1]. En cas de perturbations sur le réseau, un seul ou quelques paquets IP consécutifs seront perdus. Afin d'éviter de considérer qu'un fragment n'est pas exploitable pour une très petite fraction de données non reçues, nous mettons en œuvre la réparation des paquets perdus. En effet, la retransmission des seuls paquets permet de minimiser l'encombrement du réseau en comparaison de la retransmission d'un fragment entier ou d'une redondance FEC envoyée en permanence.

Dans les systèmes P2P actuels, le récepteur demande la retransmission de paquets perdus au pair qui lui envoie les fragments de données (c'est le cas de TCP et de la plupart des techniques de retransmission proposées avec UDP [9]). Ce pair était le meilleur pour fournir un fragment à temps, mais il n'est pas nécessairement le meilleur fournisseur des paquets à retransmettre, car ces paquets à retransmettre sont souvent en faible nombre. En effet, premièrement la capacité de traitement et d'envoi de ce premier émetteur d'une part est sûrement réduite par l'envoi des autres paquets du même segment et d'autre part peut être réduite par les sollicitations reçues par ce pair depuis qu'il a été sélectionné. Deuxièmement, la capacité de traitement et d'envoi d'un autre pair peut avoir considérablement augmentée. Troisièmement les caractéristiques en débit, délai et taux de perte peuvent varier en fonction de volume de données à transmettre. Par exemple, les petits volumes (paquet unique) subissent souvent des délais moindres. Pour ces raisons et afin d'augmenter la probabilité de recevoir à temps les paquets retransmis, nous proposons de sélectionner le pair le plus disponible pour lui demander la retransmission. Il ne sera donc pas systématiquement celui qui a envoyé initialement les paquets perdus. L'exemple présenté dans la figure 1 et la figure 2, montre un cas dans lequel ce mécanisme de retransmission est plus efficace que les mécanismes de retransmission actuels.

Dans l'exemple, le pair P2 a besoin du fragment C1 qui existe chez les pairs P1 et P3. Supposons que le RTT entre P2 et P1 (RTT (P1-P2)) est inférieur à RTT (P2-P3) et que la bande passante remontante de P3 est beaucoup plus grande que celle de P1. P1 est proche de P2, mais sa bande passante disponible ne garantit pas une réception complète du fragment avant son temps d'affichage (noté TV dans les figures 1 et 2). Pour cela, P2 a un intérêt à demander le fragment à P3. Supposons que le fragment C1 est composé des paquets (p1, p2, p3, p4, p5) et que le paquet p3 est perdu lors de son transmission entre P3 et P2. Si P2 demande la retransmission du paquet p3 à P3, le fragment complet sera reçu à l'instant T1. Cependant, si T1 est plus grand que TV (comme le cas dans la figure 1), ce fragment sera considéré non exploitable. Par contre, remarquons que si la retransmission est demandée à P1

[1] Notons que la taille des fragments dans les systèmes P2P est un domaine de recherche (voir [10]).

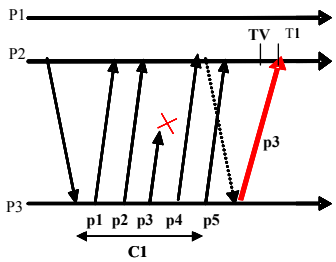

Figure 1: Exemple de transmission d'un fragment en P2P. Les retransmissions sont demandées au fournisseur initial des paquets.

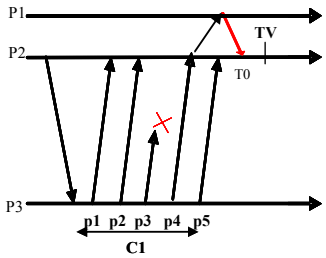

Figure 2: Les retransmissions sont demandées à un pair différent que le fournisseur initial des paquets

(figure 2), il y a une grande probabilité de recevoir le paquet retransmis à un instant T0 inférieur à TV. En effet, l'émission d'un simple paquet ne nécessite pas une grande bande passante. Ce mécanisme de retransmission peut donc permettre la réception du fragment entier avant son temps d'affichage. Ce qui améliore la qualité de la vidéo affichée.

Plusieurs métriques peuvent être utilisées pour sélectionner le meilleur fournisseur de retransmission (comme le RTT, la bande passante disponible des pairs, etc.). L'estimation de ces métriques nécessitent d'échanger périodiquement de messages de signalisations entre les pairs. Ces messages peuvent accroître l'encombrement du réseau, et ils risquent donc de contribuer à la perte de paquets. Pour limiter les messages de contrôle, nous proposons un tirage aléatoire du pair de retransmission parmi les voisins disposant du fragment affecté par la perte. Ce mécanisme présente des avantages en termes de souplesse et de robustesse. Il ne présente pas de contraintes sur le codage des données ni sur les architectures du réseau. Il ne nécessite pas de message de contrôle. La seule condition est que le récepteur ait une liste des pairs ayant le fragment affecté par la perte. Ceci est toujours possible parce que le récepteur dispose de cette liste avant de demander le fragment. Pour cela, nous supposons que ce mécanisme peut fonctionner en complément de n'importe quel système de streaming en P2P.

Pour vérifier l'efficacité de notre mécanisme, nous avons réalisé des simulations dont les résultats sont présentés dans la section suivante.

IV. LES TESTS DE PERFORMANCE

Notre mécanisme est considéré efficace s'il peut améliorer la qualité de la vidéo en comparant avec le mécanisme demandant la retransmission au pair qui a fourni initialement les paquets perdus (dans la reste de ce papier on appelle ce mécanisme : "mécanisme classique"). Un taux d'erreur assez élevé sera appliqué aux tests, afin de se placer dans des conditions ne permettant pas de corriger la totalité des pertes, mais de comparer l'efficacité des mécanismes. De fait, si les deux procédés permettaient une correction à 100%, il ne serait pas possible de les distinguer.

Pour montrer l'efficacité de ce mécanisme, nous avons développé, en utilisant l'outil de simulation OPNET Modeler, un simulateur modélisant le service de streaming en P2P. Nous expliquons dans ce paragraphe les choix effectués suivis des résultats.

Pour permettre au mécanisme de retransmission classique d'améliorer la qualité de la vidéo, il faut faire les choix suivants: (1) utiliser des pairs avec une grande capacité de bande passante remontante. (2) sélectionner toujours le pair le plus proche en terme de RTT pour lui demander les fragments. En effet, le premier choix rend le fournisseur initial des paquets plus disponible. Il a une grande capacité de bande passante, et il peut donc répondre à un grand nombre des demandes (de fragments ou de retransmissions) sans avoir de problèmes de congestions. L'utilisation du RTT pour sélectionner initialement ce pair, permet de réduire au maximum le temps de transmission des paquets vers le récepteur. En appliquant ces deux choix, la probabilité de recevoir les paquets retransmis avant le temps d'affichage de leur fragment sera augmentée. Ce qui permet d'améliorer la qualité de la vidéo.

Pour réaliser un simulateur appliquant ces choix, nous avons besoin de définir une technique de localisation des fragments. Son but est de fournir à chaque pair un ensemble d'autres pairs et les fragments dont ils disposent. Ces informations seront utilisées pour sélectionner le meilleur fournisseur d'un fragment donné. Elles seront utilisées aussi par notre mécanisme pour sélectionner aléatoirement le fournisseur de retransmission. Par contre, notre mécanisme n'impose pas des contraintes sur la technique utilisée pour avoir ces informations, et dans un système P2P il en est existe toujours une. N'importe quelle technique de localisation des fragments peut être employée. Nous avons choisi d'appliquer la technique suivante : lorsqu'un nouveau pair se connecte au système, il recevra une liste de pairs ayant la vidéo demandée. Ces pairs seront sélectionnés aléatoirement parmi les pairs ayant la vidéo. Ils ne seront pas forcément proches de ce pair. Cette sélection permet d'assurer un équilibre de charge dans le réseau. Ces pairs utilisés pour les transferts seront appelés "voisins". Le nouveau pair garde ses voisins jusqu'à la fin de la simulation. Périodiquement, il échange avec eux des messages de signalisation pour connaître les fragments disponibles chez chacun entre eux (la période d'échange est noté « period_echange »). Cette échange périodique, existe dans plusieurs systèmes P2P comme [2][3].

Le nombre des voisins (on l'appelle « nb_voisins ») ainsi que la « period_echange » sont deux paramètres importants des simulations. Ils peuvent affecter la qualité de la vidéo même s'il n'y a pas de perte sur le réseau. En effet, si le « nb_voisins » est petit, le récepteur n'aura pas beaucoup de choix pour sélectionner le meilleur fournisseur d'un fragment donné. La probabilité de choisir un pair non disponible sera donc plus élevée, avec le risque de retarder le temps de réception du fragment et d'affecter la qualité de la vidéo. En contre-partie, si le « nb_voisins » est grand, les messages de signalisation échangés entre les pairs à chaque « period _echange » peuvent générer un trafic qui peut parfois accroître l'encombrement du réseau et par suite la perte des paquets. De bonnes valeurs pour ces deux paramètres sont donc nécessaires pour garantir la qualité de la vidéo sans introduire beaucoup de trafics de contrôle dans le système. Nous avons réalisé un ensemble des simulations pour trouver ces valeurs (voir le paragraphe suivant) afin de vérifier la validité des nos algorithmes avant de tester le mécanisme de retransmission.

Notre simulateur est constitué de 500 pairs et d'un serveur central diffusant la vidéo en "Live". Les simulations ont été réalisées avec une seule vidéo (comme c'est le cas dans les littératures [12] [9]). Nous avons utilisé une vidéo de 300 Kbit/s ([12]). Les pairs sont homogènes et n'ont pas de contraintes sur leur débit descendant (comme le cas souvent, par exemple [11]). Pour respecter les choix dont on a parlé précédemment, il faut attribuer aux pairs une grande valeur de bande passante remontante. Nous avons choisi une valeur de 2 Mbit/s. Elle très grande par rapport au débit de la vidéo, ce qui permet à chaque pair de servir beaucoup des requêtes en même temps et d'assurer une bonne dissémination du contenu entre les pairs. Cette valeur est possible sur réseau FTTH, mais aussi xDSL.

A. Évaluation de la performance du modèle sans perte des paquets

L'évaluation de performance sans perte de paquets, nous permet de trouver les valeurs optimums de « nb_voisins » et « period_echange ». Ces valeurs seront utilisées après pour tester notre mécanisme avec perte des paquets.

Les métriques le plus important à mesurer sont :

- Le taux de perte des fragments (loss_fragments) : un fragment est considéré perdu si on ne reçoit pas tous ses paquets avant son temps de visualisation. Pour mesurer cette métrique, chaque pair calcule au cours de la simulation le nombre des fragments de la vidéo et le nombre des fragments considérés perdus. En utilisant les valeurs calculées par tous les pairs du système, nous calculerons à la fin de la simulation le pourcentage des fragments perdus par rapport à tous les fragments de la vidéo diffusés pendant la simulation. Ce pourcentage nous permet de connaître l'état de réception du système. S'il est de 0 %, tous les pairs ont reçu une vidéo parfaite.
- Le pourcentage de trafics de contrôle (noté « trafic_control »: c'est le pourcentage d'octets des messages de signalisation par rapport au volume totale envoyé par les pairs du système (signalisation + données). Cette métrique nous permet de voir si les trafics de contrôle gaspillent la bande passante remontante des pairs.

La figure 3 affiche l'évaluation du taux de perte des fragments en fonction du « period_echange » et « nb_voisins ». Rappelons qu'on n'a pas de perte sur le réseau. La raison de la perte des fragments est donc leur réception en retard. La figure 3 montre que le taux de perte des fragments diminue avec l'augmentation du nombre des voisins. En effet, si le nombre des voisins augmente, le récepteur a plus de choix pour sélectionner le meilleur fournisseur d'un fragment. Ceci augmente la probabilité de trouver un fournisseur disponible, et réduit la probabilité de recevoir le fragment en retard.

Cependant, la figure 3 montre aussi que le taux de perte varie aussi selon le « period_echange ». Prenons les courbes représentant les cas où la période est plus grande ou égale à 2 s. Nous remarquons que le taux de perte augmente avec l'augmentation de la période d'échange. La raison est que pour

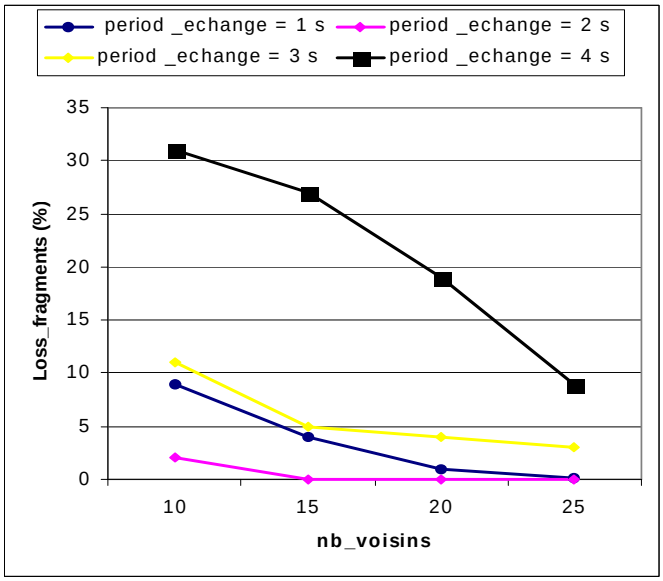

Figure 3: évaluation du taux de perte des fragments en fonction du nombre des voisins et de la période d'échange des messages de signalisations.

une grande période d'échange, les pairs doivent attendre un certain temps avant de localiser les nouveaux fragments dans le système. Ce qui peut retarder l'envoi des requêtes et par suite le temps de réception de fragments. La probabilité de recevoir les fragments en retard sera donc grande. Ce qui explique l'existence de perte des fragments avec une grande période d'échange.

Dans le cas où la période d'échange est égale à 1 s, on peut remarquer que le taux de perte n'est pas toujours à 0% (différent du cas où la période est de 2 s). Avec cette petite période, les requêtes des fragments ne seront pas envoyées en retard. La raison de la perte dans ce cas est, donc, la non-disponibilité des pairs fournisseurs. La figure 4 nous apporte une vérification. Elle évalue le pourcentage de trafics de contrôle en fonction du « nb_voisins » et «period_echang ». Rappelons que chaque pair doit échanger chaque « period_echang » des messages de contrôle avec tous ses voisins (dont le nombre est « nb_voisins »). La quantité de trafic de contrôle augmente donc avec l'augmentation du « nb_voisins ». Elle augmente aussi si la période d'échange est petite. La figure 4 montre bien cette variation de pourcentage de trafics. Elle montre aussi que si la période d'échange est de 1 s, le pourcentage de trafic de contrôle est très élevé (au moins 45%). Dans ce cas, les pairs envoient donc beaucoup de trafic de contrôle, ce qui peut les rendre moins disponibles pour envoyer les fragments qui peuvent donc arriver en retard. Ce qui explique la présence de perte lorsque la période est de 1 s.

On peut remarquer que la valeur optimum du nombre des voisins est 15 pairs, pendant que celle de la période d'échange est de 2 s. En effet, ces deux valeurs nous ont permis d'avoir une meilleur qualité de la vidéo (car le taux de perte de fragments était 0%) tout en minimisant la quantité de trafic de contrôle échangée entre les pairs. En réalisant des simulations avec ces deux valeurs, on peut s'assurer qu'on n'aura pas de pertes de fragments en raison de l'inefficacité des algorithmes implémentés (comme la localisation des fragments).

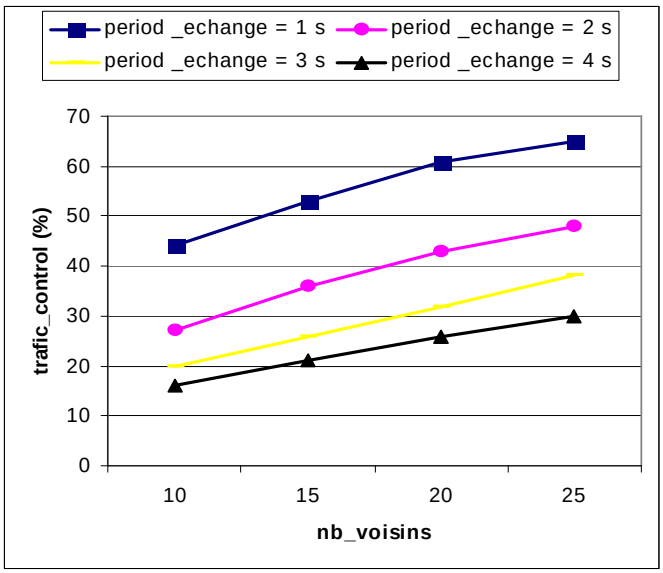

Figure 4: Le pourcentage du trafic de contrôle dans le système en fonction de nombre des voisins et de la période d'échange des messages de signalisations

B. Évaluation de la performance avec perte de paquets

Pour montrer l'efficacité de notre mécanisme, il faut le comparer avec le mécanisme de retransmission classique.

Nous avons montré dans le paragraphe précédent que notre modèle de simulation peut garantir la qualité de la vidéo s'il n'y a pas de pertes sur le réseau. Pour comparer les deux mécanismes de retransmission, nous avons introduit sur les liens du réseau une perte uniforme de 10% des paquets. Ce taux très élevé par rapport aux réseaux réels démontre l'efficacité de notre mécanisme qui s'appliquera aisément à un taux plus faible. Nous avons supposé que les messages de signalisation ne sont pas concernés par la perte, ce qui pourrait correspondre approximativement à un transport par TCP.

En utilisant les paramètres se déduisant du paragraphe précédent, nous avons réalisé deux simulations. Dans la première nous avons appliqué le mécanisme de retransmission classique sollicitant le pair émetteur du fragment, et dans la seconde nous avons appliqué le notre. Le mécanisme de retransmission le plus efficace est celui assurant la réception des paquets retransmis avant le temps d'affichage de leur fragment. Autrement dit, c'est celui minimisant le taux de perte de fragments (rappelons qu'un fragment est considéré perdu s'il est affecté par une perte ou si ses paquets arrivent en retard). Nous avons mesuré le taux de perte de fragments avec les deux mécanismes de retransmission. Les résultats sont présentés dans la figure 5.

Remarquons qu'en appliquant le mécanisme de retransmission classique, le taux de perte de fragments est 6,8 %. En revanche, notre mécanisme a minimisé ce taux à 0,9%. La plupart des fragments sont donc arrivés à temps (presque 99% de tous les fragments). Ces résultats montrent bien l'efficacité de notre mécanisme.

Les résultats montrent que le fournisseur initial des fragments n'est pas nécessairement le pair le plus adapté pour faire la retransmission en cas de perte. La demande de retransmission à ce pair n'a pas évité l'arrivée en retard de

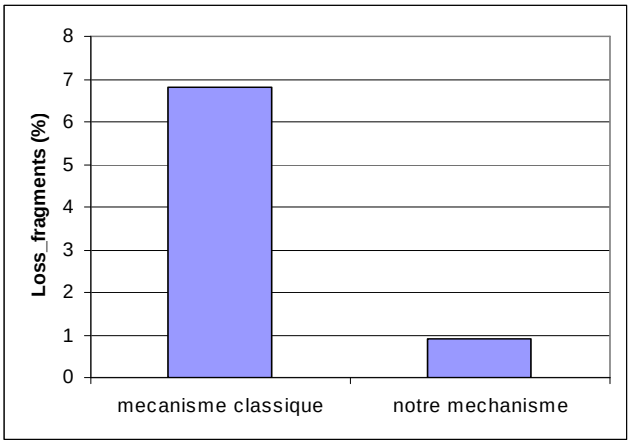

Figure 5: Comparaison entre le taux de perte des fragments dans les deux mécanismes de retransmission

fragments. Ce mécanisme de retransmission ne garantit pas donc la qualité de la vidéo. Les résultats montrent aussi que la demande de retransmission à un pair sélectionné aléatoirement parmi les voisins, peut apporter une grande amélioration de la qualité. Le taux de perte des fragments est très faible. La raison est que ce mécanisme de retransmission à augmenter la probabilité de choisir un fournisseur de retransmission disponible. Ce qui augmente la probabilité de recevoir les paquets retransmis à temps et améliore par la suite la qualité de la vidéo.

Les algorithmes et les paramètres de simulations ont été choisis pour modéliser le cas dans lequel le mécanisme classique peut bien fonctionner. L'efficacité de notre mécanisme dans ce cas, nous permet de supposer qu'il est efficace dans les autres cas.

V. CONCLUSION

La distribution de la vidéo vers un grand nombre de récepteurs est aujourd'hui un besoin fondamental. L'apparition des systèmes P2P a permis de répondre à ce besoin. Cependant, les solutions proposées aujourd'hui pour la distribution de la vidéo dans ces systèmes, souffrent d'une qualité de l'image assez médiocre. La raison principale est la perte des paquets. Cette perte est dû à la nature dynamique des nœuds participant au système, ainsi qu'au manque de garantie de performance dans le réseau IP.

Les systèmes actuels de streaming en P2P ne proposent pas de nouveaux mécanismes pour résoudre le problème de perte de paquets. Les mécanismes utilisés sont essentiellement adaptés à la correction des pertes sans contraintes particulières de temps de réparation. Nous avons proposé dans cet article un mécanisme de retransmission des paquets perdus pouvant être utilisé dans les systèmes de streaming en P2P. Il consiste à demander les paquets perdus à un pair sélectionné aléatoirement parmi les pairs ayant ces paquets. Ce pair ne sera pas systématiquement le fournisseur initial de ces paquets. Le but de ce mécanisme est d'augmenter la probabilité de choisir un fournisseur de retransmission disponible. En effet, un tel choix peut augmenter la probabilité de recevoir les paquets retransmis avant leur temps d'affichage. Ce qui permet d'améliorer la qualité de la vidéo reçue.

L'avantage de ce mécanisme de retransmission est qu'il n'impose pas de contraintes sur l'architecture P2P utilisé, ni sur le codage de données. En outre, il peut être appliqué en multicast où il est possible de faire appel à des transmissions pair-à-pair entre usagers pour récupérer localement les données perdues et limiter la congestion du serveur centrale.

Les simulations effectuées ont montré l'efficacité de ce mécanisme. Elles ont montré que le fournisseur initial des fragments n'est pas nécessairement le pair le plus adapté pour faire la retransmission en cas de perte. Ce pair ne garantit pas toujours la réception à temps des paquets retransmis, ce qui peut dégrader la qualité de la vidéo. En revanche, les simulations ont montré que la demande de retransmission à un pair sélectionné aléatoirement peut améliorer la qualité. Cependant, l'algorithme de sélection du pair fournisseur de retransmission est actuellement assez simple. Il est envisagé de proposer un algorithme optimisant la sélection des pairs avec une meilleure prise en compte des caractéristiques des réseaux.

REFERENCES

[1] F. Pianese., J. Keller, E. Biersack,"PULSE, a flexible P2P live streaming system", 9th IEEE Global Internet Symposium, Barcelona, Spain, 2006.

[2] L. Vu, I. Gupta, J. Liang, and K. Nahrstedt, "Mapping the PPLive network: Studying the impacts of media streaming on P2P overlays", Department of Computer Science, University of Illinois at Urbana-Champaign, Tech. Rep. UIUCDCS-R-2006-275, 2006.

[3] Z. Xinyan, L Jiangchuan, L. Bo, T. Shing., Y. Peter, "CoolStreaming/DONet: A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming", in IEEE Infocom, 2005.

[4] M. Castro, P. Druschel, A. M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. "Splitstream: High-bandwidth multicast in cooperative environments", International Workshop on Peer-to-Peer Systems, 2003.

[5] T. Do, K. Hua, M. Tantaoui, "P2VoD: providing fault tolerant video-on-demand streaming in peer-to-peer environment", IEEE International Conference on Communications, 2004.

[6] M. Hefeeda, A. Habib, B. Botev,D. Xu, B.Bhargava, "PROMISE: peer-to-peer media streaming using CollectCast", In Proceedings of the Eleventh ACM international Conference on Multimedia, 2003.

[7] C. Zhang, H. Jin, D. Deng, S. Yan, Q. Yuan, Z. Yin, "Anysee: Multicast-based Peer-to-Peer Media Streaming Service System", Asia-Pacific Conference on Communications, 2005.

[8] H. Luo, D. Wu, S. Ci, A. Argyriou, H. Wang, "Quality-Driven TCP Friendly Rate Control for Real-Time Video Streaming" Global Telecommunications Conference, 2008.

[9] B. Akabri.; H. Rabiee, M. Ghanbari, "Packet Loss Recovery Schemes for Peer-to-Peer Video Streaming", Third International Conference on Networking and Services. ICNS. IEEE Computer Society, Washington, 2007.

[10] N. Hegde , F. Mathieu, D. Perino," *Size Does Matter in Epidemic Live Streaming",*. Technical report 7032, INRIA, 2009.

[11] F. Picconi, L. Massoulie," Is there a future for mesh-based live video streaming ?", Eighth International Conference on Peer-to-Peer Computing, USA, 2008.

[12] M. Zhang, Q. Zhang, L. Sun, S. Yang," Understanding the Power of Pull-Based Streaming Protocol: Can We Do Better?" IEEE Journal on Selected Areas in Communications 25, 2007.

[13] H. Schulzrinne, et coll. "RFC 3550 - RTP: A Transport Protocol for Real-Time Applications", IETF standard, 2003