Étude d'un système distribué de diffusion de vidéo en direct

Frédéric Giroire, Nicolas Huin

Étude d'un système distribué de diffusion de vidéo en direct

Frederic Giroire [1] et Nicolas Huin[1]

[1] *University Nice Sophia Antipolis, I3S, UMR 7172, CNRS, INRIA, COATI, 06900 Sophia Antipolis, France.*

Dans cette étude, nous nous penchons sur les systèmes pair-à-pair pour la diffusion de vidéo en direct. Ces systèmes peuvent être divisés en deux sous-groupes. Dans les systèmes non-structurés, le choix des liens sur lesquels transmettre la vidéo n'est pas régit par une structure et la diffusion se fait de façon opportuniste. Dans les systèmes structurés, la transmission s'effectue selon un ou plusieurs arbres de diffusion. Bien que les systèmes structurés offrent une diffusion plus efficace, les systèmes non structurés sont préférés en raison de leur meilleure gestion du départ et de l'arrivée des utilisateurs (aussi appelés churn). En effet, dans les systèmes structurés, le churn casse la structure du réseau.

Dans ce papier, nous proposons plusieurs protocoles de maintenance de l'arbre de diffusion d'un système structuré soumis au churn. Nous fournissons une étude de ces protocoles par simulation ainsi qu'une analyse formelle de l'un d'entre eux. Nous donnons une estimation de métriques du système telles que le délai ou le nombre d'interruptions et temps total d'interruption de la diffusion. Nous montrons ainsi que les réseaux structurés peuvent être facilement maintenus face au churn, tout en utilisant un faible niveau d'information sur le réseau et en étant proche d'une diffusion optimale.

1 Introduction

Nous étudions les systèmes de distribution de vidéo en direct dans lesquels une source envoie du contenu en direct à plusieurs utilisateurs. Cette transmission de vidéo peut s'effectuer selon une architecture classique de clients/serveurs ou bien via une architecture pair-à-pair. Dans les systèmes pair-à-pair, contrairement à l'architecture classique, la source ne distribue pas seule la vidéo. En effet, les utilisateurs participent aussi à cette diffusion.

Les systèmes pair-à-pair peuvent être de deux types : non-structurés ou structurés. Dans les systèmes non-structurés tel que BitTorrent, les clients sont organisés selon un graphe aléatoire hautement connecté, dans lequel la diffusion du contenu se fait de manière opportuniste. Dans les systèmes structurés tel que Splitstream ou ZIGZAG [CDK$^+$03, THD03], le réseau est organisé selon un ou plusieurs arbres de diffusion. Le départ et l'arrivée d'utilisateurs dans le réseau est important dans ce type de systèmes. C'est pour cela que les systèmes non structurés sont souvent préférés. Les systèmes structurés permettent une meilleure transmission de la vidéo, mais ont une mauvaise réputation pour résister au churn.

Dans ce papier, nous montrons pourtant que ces systèmes structurés peuvent être maintenus sous churn en utilisant des protocoles assez simples. Nous proposons plusieurs protocoles de réparation de l'arbre de diffusion utilisant des niveaux d'informations différents dans la Section 2. Nous basons notre étude sur [GMNP13], où les auteurs proposent un algorithme efficace distribué de réparation de l'arbre après une ou plusieurs départs d'utilisateurs. Nous analysons un des ces protocoles de façon formel dans la Section 3 et comparons les quatre différents protocoles selon des métriques suivantes : délai maximum, nombre d'interruptions et temps d'interruption, nombre de clients sans réception dans la Section 4.

Plus de détails sur le travail peuvent être trouvés dans [GH15].

2 Modèle et opérations

Un système de diffusion est un arbre composé d'une source, située à la racine, et de n clients. Ces clients peuvent quitter et rejoindre le système mais la source est infaillible. Chaque nœud transmet la vidéo à un maximum de d fils. Un nœud possédant plus de d fils est appelé surchargé, seul d de ses fils reçoivent la

vidéo. Nous modélisons les arrivées et les départs ainsi que les réparations dans le système par des processus de Poisson de paramètres Λ et μ, respectivement.

Opérations Nous définissons deux opérations élémentaires. Chaque protocole implémente ces opérations en fonction du niveau information qu'il possède et qui est discuté ci-dessous.
- L'opération LEAVE consiste à réparer l'arbre suite au départ d'un utilisateur. Lorsqu'un nœud qui possède plusieurs fils quitte le système, un des fils adopte ses frères et se rattache à son grand-père.
- L'opération PUSH maintient l'arbre face à la surcharge d'un de ses nœuds. Comme son nom l'indique, si un nœud est saturé, un de ses fils est poussé dans un autre de ses fils.
Lorsqu'un client entre dans le système, il est rattaché à la racine.

Valeur des paramètres Puisque le temps de communication entre deux noeuds dépend de leur emplacement et de la congestion du réseau, nous modélisons le temps de réparation avec une distribution exponentiel de paramètre μ. Selon [LQK+08], le temps de communication dans ce type de systèmes est en moyenne de 79 ms. Nous considérons donc un temps moyen de réparation de 100 ms. Nous utilisons ce temps moyen comme unité de temps et normalisons le temps de réparation à 1. Une campagne d'étude de systèmes de diffusion [VGLN07] donne un temps médian de séjour des utilisateurs dans le système de 10 minutes. Cela nous donne un taux de churn normalisé Λ du système de 0.15 pour environ 1000 utilisateurs. Nous utilisons cette valeur comme valeur par défaut, mais nous considérons aussi des taux de churn entre 0 et 1. Un taux de churn de plus de 0.4 est considéré comme *haut* et un taux en-dessous de cette valeur comme bas.

Protocoles Nous proposons 4 protocoles ayant des niveaux d'informations différents. Cette information est utilisée pour choisir les nœuds impliqués dans les opérations PUSH et LEAVE, et ceux qui reçoivent la vidéo en cas de surcharge.

Dans SMALLEST SUBTREE PROTOCOL (SSP), chaque nœud connait la taille exacte de ses sous-arbres. Lors d'un LEAVE, le nœud possédant le plus grand sous-arbre devient le nouveau père. Il adopte tout ses frères et se rattache à son grand-père. Lorsqu'un nœud est surchargé, les d plus gros sous-arbres reçoivent la vidéo. Lors de l'opération de PUSH, le fils avec le troisième plus grand sous-arbre est rattaché au fils avec le deuxième plus grand sous-arbre.

Dans RANDOM PROTOCOL (RP), aucune information sur la taille des sous-arbres n'est connue. La diffusion de la vidéo s'effectue envers les nœuds les plus anciens. Le choix des nœuds s'effectue de façon aléatoire. Lors d'un PUSH, un nœud choisi aléatoirement est poussé dans un autre nœud choisi de manière aléatoire. Pour l'opération de LEAVE, le plus ancien des fils est sélectionné comme nouveau parent.

Le NO INTERRUPTION PROTOCOL (NIP) est une légère modification de RP. Son but est de diminuer le nombre d'interruptions au cours de la diffusion de la vidéo. Le protocole n'a aussi aucune information sur la taille des sous-arbres. Lors d'un PUSH, le fils qui doit être poussé en bas de l'arbre est sélectionné aléatoirement parmi les nœuds ne recevant pas la vidéo. Il est alors poussé dans un des d nœuds recevant la vidéo. Cela permet de maintenir une diffusion de la vidéo sans interruption. L'opération LEAVE est similaire à celle de RP.

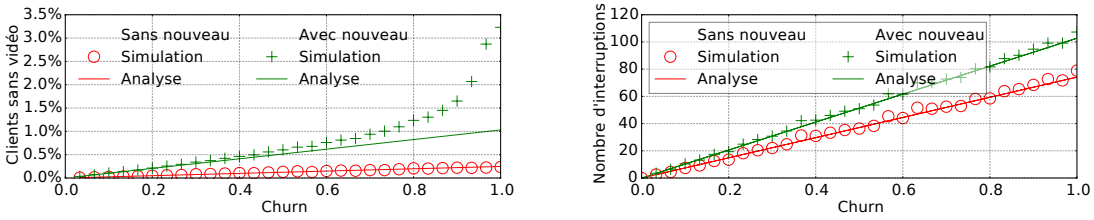
Le dernier protocole, le PARTIAL INFORMATION PROTOCOL (PIP), est une version simplifiée de SSP. Il possède une information partielle sur la taille des sous-arbres. Les nœuds peuvent être de trois types : "new", "normal" ou "big". Un nœud "new" est un nouvel arrivant dans le système. Il doit être poussé jusqu'en bas de l'arbre. Un nœud "normal" est marqué "big" lorsqu'il est adopté par un de ses frères au cours d'une opération LEAVE ou s'il est poussé durant une opération PUSH. Il ne devra pas être poussé plus bas dans l'arbre, car son sous-arbre est considéré plus gros que celui de ses frères.

3 Analyse de SSP

Nous analysons de manière formelle le temps de reconstruction de l'arbre, le temps de réparation ainsi que le nombre d'interruptions lors de l'utilisation de SSP. Nous considérons trois scénarios : un nœud quitte le système, un nœud rejoint le système et un scénario de population constante, fusion des deux premiers scénarios. Nous émettons l'hypothèse que les départs d'utilisateur sont indépendants entre eux, la réparation d'un départ se termine avant l'apparition d'un nouveau départ (cette hypothèse est réaliste vu que nous verrons que le temps moyen d'une réparation est très court et d'ordre constant, égal à $1/(2\mu)$). Pour

Scénario	Temps de réparation	Nombre de nœuds sans vidéo	Nombre d'interruptions
Départ	$\sim \frac{1}{2\mu}$	$\sim \frac{h-5}{2}\frac{\Lambda}{\mu}$	$\sim \frac{h-5}{2}t\Lambda$
Nouveau nœud	$\frac{h-2}{\mu}$	$\sim \frac{(h-2)\Lambda}{\mu}$	-
Population constante	$\frac{h-2}{\mu}$	$\sim \frac{\Lambda}{\mu}\frac{3(h-3)}{2}$	$\sim \Lambda\frac{h-3}{2}$

TABLE 1: Moyenne du temps de réparation de l'arbre, du nombre de nœuds sans la vidéo et du nombre d'interruptions en fonction de la hauteur h de l'arbre, du taux de réparation μ et du taux de churn Λ



(a) Clients sans vidéo

(b) Nombre d'interruptions

FIGURE 1: Validation des formules obtenues pour SSP en Section 3.

rappel, un départ arrive en moyenne toutes $\frac{1}{\Lambda}$ unités de temps et une opération s'effectue en moyenne toutes les $\frac{1}{\mu}$. Nous présentons une idée du calcul de ces métriques pour chaque scénario et montrons les formules obtenues dans le tableau 1. L'analyse est faite pour un niveau bas de churn.

Un nœud quitte le système. Lorsqu'un nœud quitte le système, une opération LEAVE est effectuée, suivie par un nombre d'opérations PUSH dépendant du niveau où se situait le nœud avant son départ. Si le nœud quittant le système est une feuille, l'arbre ne nécessite aucune opération pour être réparé et si ces fils sont des feuilles, seule l'opération LEAVE est effectuée. Considérons un arbre complet de taille h et définissons par i, la profondeur du nœud quittant le système. Après l'opération LEAVE, un sous-arbre de taille $2^{h-i-1}-1$ ne reçoit plus la vidéo. À chaque opération PUSH effectuée, la hauteur du sous-arbre sans réception est réduite de 1 et sa taille est divisée par 2.

Un nouveau nœud rejoint le système. Considérons l'arbre à la fin du scénario précédent. Dans ce cas, un nouveau nœud rejoint le système à la racine. La réparation de l'arbre sera complète une fois qu'il sera une feuille, après $h-2$ opérations PUSH. Pendant toute la réparation, le nœud ne reçoit pas la vidéo.

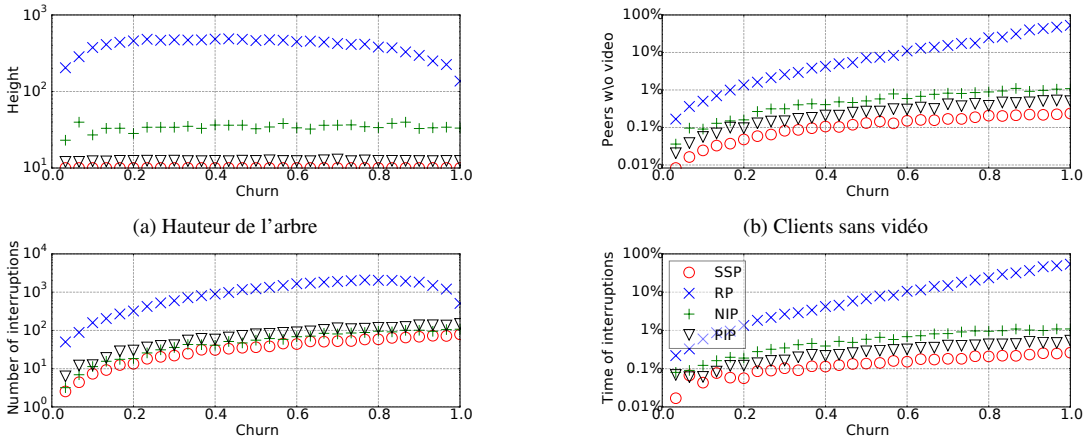
Population constante Dans ce scénario, nous considérons un arbre binaire complet de taille h où un nœud quittant le système le rejoint automatiquement. Cela correspond à une fusion des deux scénarios précédents, où nous supposons que la réparation liée au départ et à l'arrivée sont indépendantes. Le temps de réparation est le maximum des deux premiers scénarios, le nombre de nœuds sans vidéos est égal à la somme du nombre de nœuds sans vidéo des deux scénarios. Pour le nombre d'interruptions, une seule interruption s'ajoute à celle du premier scénario. Le nœud qui quitte et rejoint ne vit qu'une seule interruption.

4 Simulation

Nous simulons les quatre différents protocoles sur un système composé de 1022 clients pendant 30000 unités de temps.

Nous validons d'abord l'analyse fournie dans la Section 3. La Figure 1(a) présente la fraction du nombre de clients ne recevant pas la vidéo. Nous considérons la non-distribution de la vidéo due aux réparations ainsi que celle due aux arrivées dans le système. L'analyse pour la réparation effectuée pour une valeur de churn bas est validée par les simulations pour le nombre de clients sans vidéo et pour le nombre d'interruptions, Figure 1(b). Nous voyons qu'elle est même valide dans le cas d'un churn *haut*, sauf pour le nombre de clients sans vidéo quand on considère des nouveaux arrivants dans le système. En effet, dans ce cas, la racine devient un goulot d'étranglement car elle ne peut plus effectuer d'opérations PUSH assez rapidement. Cela ne rajoute cependant pas d'interruptions dans le système.

Dans la figure 2, nous comparons le comportement des quatre protocoles selon quatre métriques principales : (i) la hauteur de l'arbre, (ii) le nombre de clients sans la vidéo, (iii) le nombre d'interruptions et (iv)



(a) Hauteur de l'arbre

(b) Clients sans vidéo

(c) Nombre d'interruptions

(d) Temps d'interruption

FIGURE 2: Comparaison des protocoles sur un système de 1022 clients pendant 30 000 unités de temps.

le temps d'interruption de la diffusion de la vidéo. Nous voyons que RP n'est pas utilisable en pratique. En effet, l'arbre atteint jusqu'à une hauteur de 500. La raison en est que, puisque la sélection des nœuds qui sont poussés lors d'un PUSH est aléatoire, il est possible de pousser un gros sous-arbre ce qui va alors déséquilibrer fortement l'arbre de diffusion. De plus, lorsque le churn est haut, plus de la moitié des clients ne reçoivent pas la vidéo et une interruption est subie environ toutes les 1,5 secondes. Par contre, malgré une connaissance aussi nulle du réseau, NIP se comporte bien mieux que RP. La taille de l'arbre ne dépasse pas les 40, même pour des valeurs hautes de churn. Sans surprise, SSP montre les meilleurs résultats. La taille maximum de l'arbre de 10, ce qui est optimal, le nombre d'interruptions et le temps d'interruption sont faibles (75 interruptions et environ 0.25% de temps d'interruption pour un churn de 1). Finalement, nous montrons que connaitre la taille exacte des sous-arbres n'est pas vraiment nécessaire. Une faible quantité d'information est en effet suffisante à PIP pour montrer des résultats proches de SSP.

5 Conclusion

Dans ce papier, nous étudions les systèmes pair-à-pair de distribution de vidéo en direct, basé sur des réseaux structurés. Nous proposons quatre protocoles utilisant des niveaux d'informations différents qui implémentent des opérations simples de réparation de l'arbre. Nous effectuons l'analyse d'un de ces protocoles ainsi que des simulations numériques des protocoles qui valident la précédente analyse. Nous montrons que les *réseaux structurés peuvent être facilement maintenus face au churn, tout en utilisant un faible niveau d'information sur le réseau et en étant proche d'une diffusion optimale*.

Références

[CDK+03] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. SplitStream : high-bandwidth multicast in cooperative environments. In *ACM symposium on Operating systems principles*, 2003.

[GH15] F. Giroire and N. Huin. Study of repair protocols for live video streaming distributed systems. In *IEEE GLOBECOM*, 2015.

[GMNP13] F. Giroire, R. Modrzejewski, N. Nisse, and S. Pérennes. Maintaining balanced trees for structured distributed streaming systems. In *SIROCCO*, volume LNCS 8179, pages 177–188, Italy, 2013.

[LQK+08] B. Li, Y. Qu, Y. Keung, S. Xie, C. Lin, J. Liu, and X. Zhang. Inside the new coolstreaming : Principles, measurements and performance implications. In *27th IEEE ICC*, 2008.

[THD03] Duc A Tran, Kien A Hua, and Tai Do. Zigzag : An efficient peer-to-peer scheme for media streaming. In *IEEE INFOCOM 2003*, volume 2, pages 1283–1292, 2003.

[VGLN07] L. Vu, I. Gupta, J. Liang, and K. Nahrstedt. Measurement of a large-scale overlay for multimedia streaming. In *HPDC*, pages 241–242. ACM, 2007.