

# Une architecture hybride Client/Serveur et Pair-à-Pair pour le streaming vidéo sur l'Internet

Nassima Bouzakaria, Majd Ghareeb, Benoît Parrein  
LUNAM Université, Université de Nantes,  
IRCCyN UMR CNRS 6597, Polytech Nantes, France  
Email : prénom.nom@polytech.univ-nantes.fr

Soufiane Rouibia  
TMG  
Saint Sébastien-sur-Loire, France  
Email : rouibia@tmg.eu

**Résumé**—Dans l'objectif de fournir un système efficace de streaming vidéo sur Internet, nous proposons dans cet article d'étendre la technique Dynamic Adaptive Streaming over HTTP (DASH) vers le mode de communication Pair-à-Pair (P2P). DASH est généralement mis en œuvre dans un modèle classique client/serveur. Notre proposition est d'appliquer cette technique sur une architecture hybride qui est appelée P2PWeb combinant les architectures réseaux client/serveur et P2P pour permettre la disponibilité du service vidéo même en cas de saturation du serveur. Une expérimentation est réalisée sur une plate-forme externalisée de 23 machines virtuelles dans l'Internet public. Les résultats en termes de QoS nous permettent de valider notre approche hybride pour le service visé. La proposition de regroupement des segments vidéos semble améliorer les délais de transfert de bout-en-bout.

**Index Terms**—Streaming vidéo ; Architectures réseaux hybrides ; DASH ; P2PWeb.

## I. INTRODUCTION

Ces dernières décennies, les technologies Internet ont été témoins d'une révolution énorme, accompagnée de phénomènes de la diminution des coûts de stockage sur disque, le développement de l'accès haut débit résidentiel et le déploiement de nouveaux mécanismes orientés vers la qualité de service. Tous ces facteurs ont permis au streaming vidéo d'occuper une partie considérable des communications Internet.

En générale, le streaming en temps réel se fait via les protocoles RTP/UDP en raison de sa faible latence. Toutefois, une tendance récente dans la littérature montre plusieurs avantages de l'utilisation du protocole HTTP pour le streaming vidéo à la demande et en directe. Ce mode simple de transport est appelé Dynamic Adaptive Streaming over HTTP (DASH). Il présente un intérêt substantiel en matière de QoS [1], [2], [3] à tel que point que l'offre industrielle OTT (Over-The-Top) qui s'appuie sur le DASH devrait supplanter l'offre IPTV standard (principalement basée sur UDP) dans les prochains semestres.

Cependant, pour répondre aux exigences particulières de streaming vidéo, le mode de communication traditionnel client/serveur (C/S) présente plusieurs inconvénients. De là, le développement de solutions alternatives telles que le Peer-to-Peer (P2P) est devenu une nécessité.

Mais pour fournir un service de streaming vidéo efficace, ni l'architecture client/serveur, ni le P2P ne peut atteindre seule l'objectif. Le serveur centralisé de l'architecture client/serveur ne peut pas supporter les énormes charges sur l'Internet, ce qui

compromet l'évolutivité du système et conduit à une mauvaise qualité de la vidéo. Il requiert des fonctionnalités d'accessibilité élevées via les CDN (*Content Delivery Network*) notamment qui ne peuvent pas être fournies facilement à un prix avantageux [4]. D'autre part, l'architecture P2P a besoin d'un nombre suffisant de sources de contenu pour le service de streaming. En outre, il n'est pas facile de déterminer la contribution de chaque pair, qui peut aussi affecter la fiabilité du système. Pour faire face à ces questions, différentes études de recherche proposent des architectures de réseaux hybrides avec un compromis entre les architectures client/serveur et P2P [4], [5], [6], [7].

Dans ce cadre, une architecture hybride prometteuse nommée P2PWeb a été récemment proposée. P2PWeb aide à intégrer les technologies P2P dans un environnement Web afin de fournir un service efficace de livraison du contenu sur l'Internet tel que présenté dans [8], [9]. Ces deux travaux présentent l'intérêt de l'architecture P2PWeb vis-à-vis de l'architecture classique client/serveur en termes de QoS respectivement dans un contexte de simulation de réseaux de paquets et de déploiement de machines virtuelles.

L'objectif principal de ce travail présenté dans cet article consiste à démontrer l'efficacité de l'architecture P2PWeb pour le streaming vidéo sur Internet en termes de QoS et QoE (Quality of Experience). Pour cet objectif, nous proposons de déployer la technique DASH (Dynamic Adaptive Streaming over HTTP) qui est généralement utilisée en mode client/serveur vers le réseau P2P du protocole P2PWeb.

Nous abordons cet article par la présentation de la technique DASH en section II. Puis, nous présentons l'architecture hybride P2PWeb ainsi que ses protocoles réseaux en section III. Ensuite la section IV illustre notre contribution pour étendre le DASH vers l'architecture P2PWeb. Enfin, en section V nous montrons quelques évaluations de performance et les résultats des expérimentations obtenues au cours de la mise en œuvre de cette contribution. Nous terminons par une conclusion et quelques perspectives en section VI de manière à étendre cette première expérimentation.

## II. DYNAMIC ADAPTIVE STREAMING OVER HTTP (DASH)

Le streaming sur HTTP suscite, à l'heure actuelle, l'intérêt de nombreux travaux dans la littérature [2], [10]. L'utilisation

du protocole HTTP permettrait de traverser facilement les pare-feux et les routeurs du réseau, en utilisant le port ouvert pour l'accès à Internet, ce qui n'est pas évident pour une transmission sur RTP/UDP. Il permet aussi de réutiliser les infrastructures existantes de réseau, telles que les caches du réseau qui sont largement déployés, ce qui réduit la quantité de trafic sortant que les serveurs doivent supporter et donc d'éviter les problèmes d'extensibilité (passage à l'échelle). Le streaming sur HTTP offre une fiabilité et une simplicité de déploiement car le protocole HTTP et TCP/IP sont largement déployés dans le monde internet. En plus, l'implémentation d'un système de streaming sur HTTP est simple à réaliser, il suffit d'utiliser des serveurs web pour la distribution de contenu multimédia. En outre, ils n'ont pas besoin d'une fonctionnalité spécifique pour faire face à ces données [2], [3], [10].

Le téléchargement progressif sur HTTP, généralement utilisé pour la VoD (Video on Demand) se différencie du téléchargement simple pour lequel le client doit attendre d'avoir complètement téléchargé le fichier pour pouvoir ensuite le lire depuis son espace de stockage local. Le téléchargement progressif permet à l'utilisateur de lire le média alors que le fichier est encore en cours de téléchargement, ce qui donne un aspect de streaming au client [2]. Cette méthode de livraison utilise à la base un simple serveur web. Le transport de la vidéo s'effectue par l'intermédiaire de plusieurs chunks HTTP qui vont être téléchargés un par un [1], [2].

DASH est considéré comme étant un nouveau standard de diffusion vidéo en streaming. Ainsi, le nouveau format de diffusion permettra d'adapter la qualité d'affichage en fonction du débit Internet disponible. Le fichier multimédia dans son ensemble est découpé en petites parties, appelées segments qui sont les ressources affectées à un serveur et désignés par des URL pour le téléchargement (éventuellement avec un supplément de paramètre HTTP pour le *byte range* par exemple). Cette méthode prend en charge la vidéo à la demande et le streaming en direct.

DASH définit deux formats permettant la livraison de contenu vidéo pour les clients, il s'agit du format du média à télécharger et la description du média à télécharger (MPD) [2], [3]. Le Multimedia Presentation Description (MPD) est un document XML qui décrit les différents médias disponibles sur le serveur ainsi que leurs localisations. Le client peut sélectionner, via toutes les métadonnées fournies, le média qui répond à ses exigences. Celui-ci s'adapte aux propriétés du réseau du client [3]. Le MPD se compose d'une ou de plusieurs périodes où une période peut être constituée de plusieurs éléments média. Ces derniers peuvent exister en différentes versions, par exemple différent débit binaire ou de résolutions, etc. En DASH, une telle «version» de flux est appelée une représentation, et toutes les représentations sont regroupées par la suite dans un Adaptation Set. Chaque représentation comprend un ou plusieurs segments. Les segments sont des morceaux qui contiennent les données des médias dans une

séquence temporelle. Chaque segment a une URL, c'est-à-dire un emplacement adressable sur un serveur qui peut être téléchargé via HTTP GET ou HTTP GET avec des plages d'octets utilisées (Byte Range Mode) afin d'accéder à une partie spécifique du segment [1].

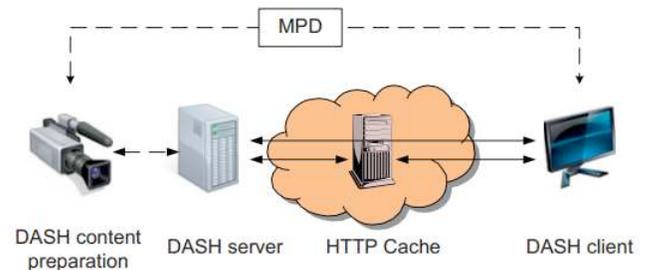


FIGURE 1. Exemple d'une architecture DASH (extraite de [2]).

L'architecture DASH représentée dans la figure 1 est constituée :

- d'un composant de préparation de contenu DASH qui s'occupe de la préparation des segments ;
- du document MPD ;
- d'un serveur DASH qui est un simple serveur web dans lequel les segments DASH sont stockés. Le client peut y accéder pour télécharger les segments et peut être aussi le MPD ;
- d'un cache HTTP qui peut être utile, afin d'alléger la charge sur le serveur ;
- d'un client DASH qui récupère les segments et effectue les opérations nécessaires pour que le contenu soit acheminé à l'utilisateur.

### III. L'ARCHITECTURE HYBRIDE P2PWEB

Comme nous l'avons déjà présenté, chacune des architectures client/serveur et P2P a ses propres limites pour les applications de streaming vidéo. Pour cette raison, l'architecture de réseau hybride a été proposée afin de bénéficier des avantages des deux architectures et de surmonter leurs inconvénients. Dans certains travaux, les auteurs proposent de libérer le serveur après un instant donné et de passer à un mode de communication P2P pur [4], [7]. D'autres approches proposent de faire le streaming dans un mode P2P, puis de re-diriger les demandes des clients vers le serveur de vidéo en cas de besoin [5]. L'architecture hybride P2PWeb traite un cas d'utilisation différent.

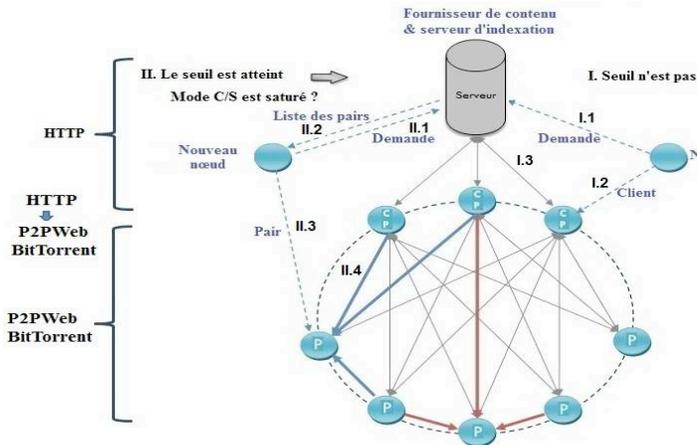


FIGURE 2. L'architecture hybride P2PWeb et les protocoles réseaux utilisés.

En P2PWeb, le serveur joue deux rôles : un serveur de contenu média et un serveur d'indexation P2P. Le serveur continue à répondre aux demandes des clients tant qu'il n'est pas surchargé. Puis, quand il arrive à un seuil de saturation, il commence à rediriger les nouveaux clients (pairs) pour récupérer le contenu à partir des autres clients / pairs, qui sont déjà présents dans le réseau et ayant reçu le contenu demandé. P2PWeb vise à réduire la charge sur le serveur et fournir une meilleure QoS (Quality of Service) et QoE (Quality of Experience).

Dans cette section, nous introduisons d'abord l'architecture hybride P2PWeb avec plus de détails, ensuite nous présentons les protocoles réseaux utilisés par cette architecture.

### A. L'architecture du système

La figure 2 résume le modèle de l'architecture du réseau hybride de manière concise et graphique. Lors de l'apparition d'un nouveau nœud qui demande un contenu proposé par un serveur web, ce dernier lui transmet la ressource désirée en mode client/serveur comme le ferait le protocole HTTP normalement. Mais une fois que le serveur commence à être saturé par le nombre de requêtes, il devient un serveur d'indexation (*tracker*) des fichiers et lui répond par une liste des utilisateurs ayant déjà téléchargé ce contenu et ayant accepté de le partager en utilisant le protocole P2PWeb BitTorrent (voir partie B). Autrement dit, le contenu va être récupéré par le nœud en mode pair-à-pair, qui permet d'alléger la charge du serveur tout en fournissant une meilleure qualité de service (QoS), une qualité d'usage (QoE) et de la disponibilité du contenu à l'utilisateur.

### B. Les protocoles réseaux utilisés dans P2PWeb

P2PWeb utilise le protocole HTTP en mode client/serveur et le protocole P2PWeb BitTorrent en mode pair à pair comme le montre la figure 2. Le protocole P2PWeb BitTorrent est

une version modifiée du code source BitTorrent, développé en langage Python. Les modifications appliquées consistent à adapter les requêtes pour les échanges sur le Web. Les premiers morceaux à télécharger sont organisés afin d'avoir des morceaux séquentiels et non aléatoires. Le délai de transfert de bout-en-bout est utilisé pour mesurer la qualité de service (QoS). Les clients de P2PWeb ne sont pas des clients ordinaires car ils reçoivent les morceaux directement du serveur, et à un moment donné, ils deviennent des fournisseurs de ces morceaux.

## IV. EXTENSION DE DASH SUR L'ARCHITECTURE P2PWEB

Le DASH (Dynamic Adaptive Streaming over HTTP) est initialement conçu pour des architectures classiques Client/Serveur. Or, les réseaux P2P jouent un rôle prépondérant dans la diffusion de contenu. Nous proposons donc ici d'étendre le protocole DASH à une architecture P2P ainsi qu'à notre architecture hybride P2PWeb. Dans ce qui suit, nous présentons le dispositif qui nous a permis de faire un premier déploiement du DASH sur le protocole BitTorrent.

Un composant de préparation de données DASH s'occupe de la préparation du contenu vidéo en le découpant en plusieurs segments ainsi que la création du document descriptif MPD (Media Presentation Description). Tous ces éléments vont être stockés dans le serveur Web. Les clients accèdent dans un premier temps au serveur afin de récupérer le MPD. À ce moment, ils peuvent sélectionner la représentation (la version) de la vidéo qui répond à leurs souhaits et qui s'adapte aussi à leurs capacités réseaux. Ensuite, les clients commencent à télécharger les segments désignés par des URL à partir du serveur tant qu'il n'est pas saturé.

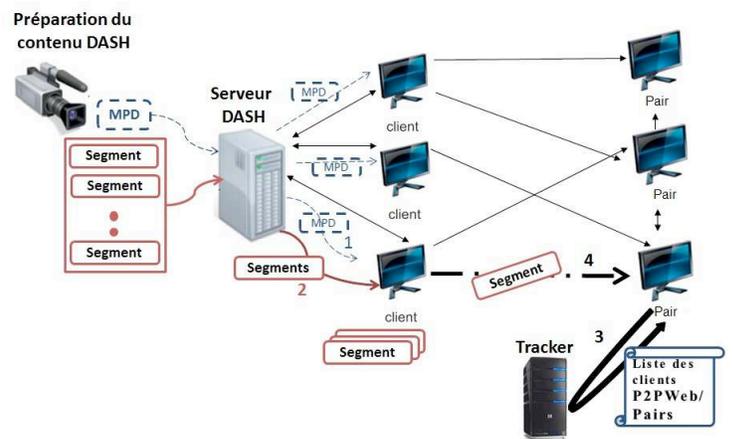


FIGURE 3. Déploiement de DASH sur l'architecture hybride P2PWeb.

En outre, le serveur déclare au *tracker* chaque nouveau client en tant que propriétaire d'un nouveau contenu. Malgré que les clients P2PWeb reçoivent le contenu directement depuis le serveur, nous devons viser à sa retransmission potentielle. En effet, ils reçoivent la vidéo en termes de segments et

à un moment donné, ils deviennent des fournisseurs de cette vidéo.

Lorsque le serveur arrive à un seuil de saturation, il ne sera pas en mesure de livrer le contenu à d'autres clients. Le *tracker* prend la responsabilité de répondre aux nouvelles demandes, cela se fera par l'envoi d'une liste des clients P2PWeb/pairs, qui ont déjà téléchargé et ayant accepté de partager le contenu désiré. Ainsi, au lieu de télécharger les segments de la vidéo directement depuis le serveur, les pairs récupèrent leurs segments demandés à partir des clients P2PWeb/pairs. Le téléchargement des segments se fait progressivement c'est-à-dire le pair commence à lire dès qu'il reçoit un segment, en attendant l'arrivée des autres sans attendre la réception de la totalité. Alors, il peut directement commencer à partager ce qu'il a téléchargé avec les autres pairs en utilisant le protocole P2PWeb BitTorrent. La figure 3 explique la démarche et les différentes interactions effectuées.

## V. EVALUATION DES PERFORMANCES ET RESULTATS D'EXPERIMENTATION

Cette section est articulée autour de l'environnement de travail et les scénarios d'expérimentation qui ont permis l'application du DASH sur l'architecture P2PWeb. L'expérimentation concerne le transport des segments DASH au sein du réseau P2P de l'architecture P2PWeb. Cette expérimentation s'inscrit dans le prolongement de celles menées pour comparer l'architecture P2PWeb à l'architecture Client/Serveur classique dans un cadre de simulation [8] ainsi que dans un cadre de déploiement de machines virtuelles [9]. Nous présentons les résultats obtenus liés principalement à la vitesse de téléchargement d'un fichier vidéo comme critère de Qualité de Service (QoS) en supposant la réception de la totalité de la vidéo.

### A. L'environnement de travail

L'environnement de travail qui a servi pour les expérimentations effectuées est basé sur une plate-forme externalisée qui regroupe un serveur Web, 23 machines virtuelles d'adresse IP publiques, MP4Box comme étant un composant de préparation de contenu DASH et un protocole P2PWeb BitTorrent basé sur le Mainline (3.9.1) de BitTorrent. Le nombre de clients P2PWeb est fixé dans cette étude à 8 contre 15 pairs (dont 1 pair sur lequel s'effectue les mesures de QoS) soit une répartition égale de clients et de pairs au sein du réseau hybride expérimental P2PWeb. La méthode d'injection des segments au sein du réseau hybride ainsi que le débit de téléchargement d'une liaison client-pair va nous permettre de définir des scénarios distincts (le scénario 1 et le scénario 2 dans la suite). En termes de QoS, seul le temps de téléchargement est considéré dans cette étude. Le critère fondamental est la réception de l'intégralité des segments.

Le fichier vidéo source utilisé dans l'expérimentation est une vidéo d'une durée de 2 minutes et 30 secondes encodée au format MP4 « ISO Base Media File » d'une capacité totale de 40,6 Mo. Au niveau du protocole DASH, la vidéo MP4 est fragmentée en 15 segments dont la taille avoisine les 3

Mo par segment. Chaque segment correspond à 10 secondes de vidéos. Le fichier MPD (*Media Presentation Description*) au format XML englobe tous les détails de cette segmentation.

B. Présentation des scénarios et résultats d'expérimentation  
Nous distinguons ici 2 scénarios en fonction de l'allocation du nombre de segments DASH aux fichiers *torrent* propres au réseau BitTorrent utilisé au sein du protocole P2PWeb ainsi que la mise en ligne de ces fichiers *torrent* au cours du temps. Le débit de liaison entre client et pair varie entre les scénarios de manière à prendre en compte la répartition des données et de la charge au sein du réseau P2P.

#### 1) Scénario 1 : streaming de segments un par un

Dans ce scénario, nous faisons la correspondance directe entre un segment DASH et un fichier *torrent*. Ces fichiers *torrent* sont disponibles progressivement au sein du protocole P2PWeb dans l'ordre des segments de la vidéo avec un délai inter-segment de 5 secondes et ceci de manière à ne pas saturer le réseau P2P. De façon empirique, nous avons fixé le débit des liens descendant à 30 Ko/s pour parvenir au téléchargement intégral de la séquence vidéo sur le pair de mesure. Par expérience, un débit supérieur (60 Ko/s) comme inférieur (15 Ko/s) à ce débit empirique ne permet pas d'aboutir au téléchargement complet. Nous supposons dans un cas une saturation des liens dû au nombre de segment et dans l'autre cas, un débit simplement trop faible pour le volume de données à transmettre.

La figure 4 représente le pourcentage du contenu vidéo fragmenté en 15 segments et téléchargé par la machine de mesures. Nous pouvons remarquer qu'avec 8 clients P2PWeb, 14 pairs et un débit de téléchargement de 30 Ko/s pour chaque liaison client/pair et pair/pair, le pair de mesure a eu besoin de 180 secondes pour télécharger la totalité des segments constituant la vidéo.

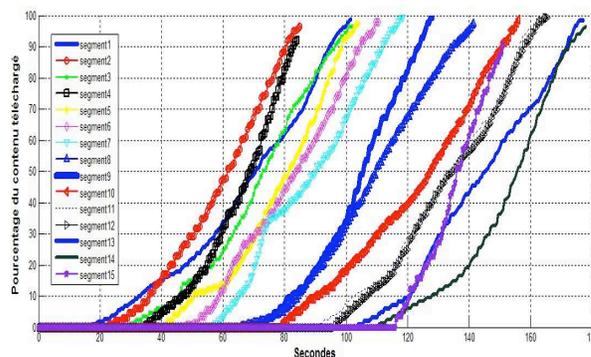


FIGURE 4. Pourcentages de l'ensemble des segments téléchargés sur le pair de mesures.

Le résultat obtenu n'est pas satisfaisant pour le service vidéo streaming visé. Ceci nous conduit à proposer un regroupement des segments dans un scénario 2 afin d'obtenir un temps de téléchargement ne dépassant pas les 60 secondes.

#### 2) Scénario 2 : streaming des groupes des segments

Dans ce scénario, la population de clients P2PWeb et de pairs est identique au scénario 1. Par contre les clients P2PWeb/pairs qui sont déjà connectés au réseau n'envoient pas les segments séparément un par un mais sous forme de groupes de segments alloués à un *torrent*. Chaque *torrent* regroupe maintenant 5 segments DASH donnant ainsi 3 groupes dont chacun a une durée de 50 secondes. Les clients P2PWeb/pairs envoient les groupes constituant la vidéo un par un aux pairs en fonction de leurs requêtes avec un débit de téléchargement maximum de 150 Ko/s pour chaque pair et un délai de 25 secondes entre la transmission d'un groupe et d'un autre. Ce débit est largement supérieur au scénario 1. Il est possible de l'affecter de part le regroupement des segments au sein d'un plus petit nombre de *torrents*.

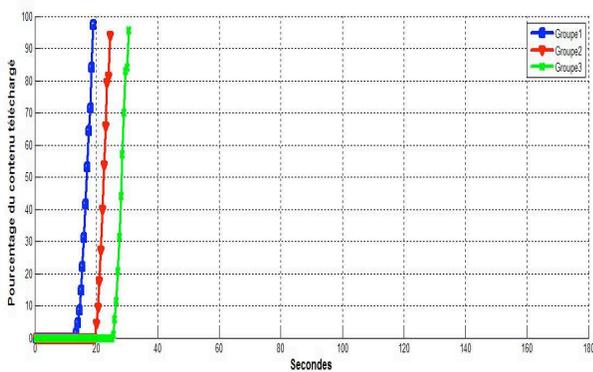


FIGURE 5. Pourcentages téléchargés sur le pair de mesure par groupe de segments en fonction du temps.

La figure 5 représente le pourcentage des 3 groupes téléchargés un par un par la machine de mesure dans une durée de 30,5 secondes. Nous en déduisons une progression assez rapide et remarquable dans le téléchargement des groupes constituant la vidéo.

### 3) Comparaison et discussion

A l'issue des résultats obtenus au niveau des deux scénarios, nous avons effectué une analyse de progression du pourcentage de contenu téléchargé par la machine de mesures en termes de délai de téléchargement. Un calcul de moyenne a été réalisé sur des ensembles de 5 segments chacun appartenant au premier scénario dans le but de rendre la figure 4 du scénario 1 plus lisible. La figure 6 nous permet de comparer simplement les 2 scénarios. Le DASH avec un groupement de segments semble plus performant que le DASH avec un transfert segment par segment en termes de délai de transfert de bout-en-bout. Une expérimentation plus complète est cependant nécessaire pour effectuer des préconisations voire une spécification complète.

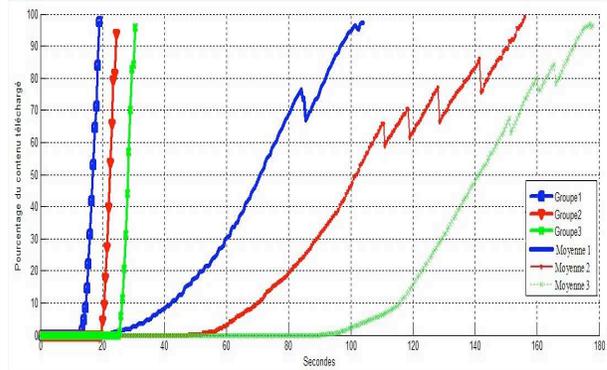


FIGURE 6. Comparaison du pourcentage du contenu téléchargé en termes de segments et groupes de segments DASH.

## VI. CONCLUSION

Dans cet article, notre objectif est de développer un système efficace de streaming vidéo sur l'Internet avec comme hypothèse réaliste (mais discutable) que le protocole de transport est simplement HTTP. La discussion sur l'intérêt du protocole HTTP vis-à-vis des protocoles normaux de transport type RTP/UDP est en dehors du champs d'investigation de ce travail. Nous proposons ici de supporter le protocole DASH (*Dynamic Adaptive Streaming over HTTP*) qui est généralement transmis en mode client/serveur vers une architecture hybride client/serveur et P2P appelée P2PWeb. Nous démontrons la faisabilité de cette approche par le développement d'une plate-forme d'expérimentation constituée d'une vingtaine de machines virtuelles avec une population mixte de clients et de pairs.

L'exploitation de cette plate-forme a permis d'expérimenter un premier paramétrage basé essentiellement sur l'allocation des segments DASH aux fichiers *torrents*. Il semble que le groupement de segments au sein d'un seul torrent permet d'augmenter les débits de téléchargement tout en assurant son intégralité. Il demeure néanmoins un grand champs d'expérimentations possibles notamment par le passage à l'échelle, des populations variables de clients et de pairs ainsi que pour le service visé des encodages et des durées de séquences plus représentatives.

## REMERCIEMENT

Le travail présenté dans cet article s'insère dans le projet P2PWeb, qui est soutenu par l'OSEO et les régions de Bretagne et Pays de la Loire dans le cadre du pôle de compétitivité « Images et Réseaux ». Les auteurs tiennent également à remercier les correcteurs de cet article pour leurs remarques pertinentes.

## RÉFÉRENCES

- [1] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE Multimedia*, vol. 18, pp. 62–67, 2011.
- [2] Y. Sanchez, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. D. Vleeschauwer, W. V. Leekwijck, and Y. L. Louédec, "Efficient http-based streaming using scalable video coding," *Image Communication*, vol. 27, pp. 329–342, 2012.

- [3] T. Stockhammer, "Dynamic adaptive streaming over http-standards and design principles," in *Proceedings of the second annual ACM conference on Multimedia systems*, pp. 133–144, ACM, 2011.
- [4] D. Xu, S. S. Kulkarni, C. Rosenberg, and H. Keung Chai, "A cdn-p2p hybrid architecture for cost-effective streaming media distribution," *Computer Networks*, vol. 44, pp. 353–382, 2004.
- [5] A. Bakker, R. Petrocco, J. Gerber, V. Grishchenko, D. Rabaïoli, and J. Name, "Online video using bittorrent and html5 applied to wikipedia," *Int. Conf. on Peer-to-Peer Computing*, pp. 1–2, 2010.
- [6] R. Mattson, *Enhancing HTTP to improve page and object retrieval time with congested networks*. PhD thesis, La Trobe University, Melbourne, Australia, 2008.
- [7] Y.-C. Tu, J. Sun, M. Hefeeda, and S. Prabhakar, "An analytical study of peer-to-peer media streaming systems," *ACM Trans. Multimedia Comput. Commun. App.*, vol. 1, pp. 354–376, 2005.
- [8] B. Parrein, N. Normand, M. Ghareeb, G. d'Ippolito, and F. Battisti, "Finite radon coding for content delivery over hybrid client/server and p2p architecture," *5th International Symposium on Communications, Control and Signal Processing*, pp. 1–4, Mai 2012.
- [9] S. Rouibia, M. Ghareeb, B. Parrein, M. Biazzi, R. Carvajal-Gomez, A. Perez-Espinosa, and P. Serrano-Alvarado, "Towards a hybrid client/server and p2p architecture for content delivery over the internet," *Accepté à CFIP 2012*, Bayonne, Octobre, 2012.
- [10] Y. S. de la Fuente, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. D. Vleeschauwer, W. V. Leekwijck, and Y. L. Louédec, "idash :improved dynamic adaptive streaming over http using scalable video coding," *MMSys'11 Proceedings of the second annual ACM conference on Multimedia systems*, pp. 257–264, 2011.