

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université SAAD DAHLAB de BLIDA 1

Faculté des sciences

Département d'informatique

Domaine : informatique

Filière : informatique



SPÉCIALITÉ : SYSTÈMES INFORMATIQUES ET RÉSEAUX

MÉMOIRE DE MASTER

Amélioration de la qualité d'expérience dans les services de vidéo streaming sur internet.

Présenté par : TALAILEF AHLEM & ZOUBIRI RANIA

Promoteur : Douga Yassine

Co-promoteur : Souane Naima

Devant le jury :

Mme S.AROUSSI

Présidente

Mme MANCER

Examinatrice

Année Universitaire : 2019-2020

Remerciements

Qu'il nous soit permis de remercier

Dieu le tout Puissant

Pour que tous nos efforts soient fructueux et bénéfiques

*Nous tenons à exprimer toute notre reconnaissance à notre encadrant et rapporteur de mémoire monsieur «**DOUGA Yassine**» : nous le remercions vivement de nous avoir encadrées, orientées, aidées et conseillées.*

Nous adressons également nos sincères remerciements à tous les professeurs, intervenants pour leur disponibilité durant notre cursus universitaire.

Merci à toute personne qui par ses paroles, ses écrits, ses conseils ainsi que ses critiques a guidé nos réflexions.

Ahlem & Rania

Dédicace

Mes remerciements vont tout d'abord à ALLAH tout puissant pour m'avoir permis de poursuivre mes études dans de bonnes conditions et de faire de moi ce que je suis aujourd'hui.

Je le dédie à :

Ma très chère Mère, qui a toujours été présente pour moi, dans les moments les plus difficiles et qui sans cesse veille sur moi avec ses prières, pour ses grands sacrifices et tout l'amour qu'elle me porte.

Mon très cher Père, pour tous ses conseils et pour toute la confiance qu'il a mise en moi et pour son dévouement pour mon bonheur.

Que dieu me les garde.

Meschers frères: Mohamed & Billy & Mohamed CH & Khiredline

Mes chères sœurs : Nana & Habiba & Nacima & Sara

Les princes et princesses : Hadil, Malek, Sadek, Nada, Louai, Ranim, Amina, Roufane, Lokman, Diyaa, Mohamed ...

Toute la famille : TALAILEF & ALIOUA

Tous mes amis(es), plus particulièrement « Meriem, Ryma, Sara, Aicha, Fella..... » avec qui j'ai passé de très bons moments.

Ma binôme ZOUBIRI RANJA et toute sa famille

TALAILEF AHLEM

Dédicace

Je dédie ce modeste travail à :

Ma très chère maman : qu'elle trouve ici l'hommage de ma gratitude qui si grande qu'elle puisse être ne sera à la hauteur de ses sacrifices et ses prières pour moi.

A celui qui a toujours garni mes chemins avec force et lumière, mon premier encadrant dès ma naissance... Mon très cher papa.

A ma sœur Naila et mes frères Anis et Ali and ma cousine Linda

A toutes ma famille

A mon binôme Ahlem

A tous ceux que j'aime et qui m'aiment qu'ils trouvent ici l'expression de mes sentiments les plus dévoués et mes vœux les plus sincères :

Que Dieu le tout puissant vous préserve tous et vous procure sagesse et bonheur.

Rania

ملخص

اليوم أصبح البث وسيلة أساسية لنشر المعلومات للجمهور ومع الانتشار الواسع لخدماته مثل YouTube و Netflix وغيرها، فإنه يستخدم نسبة كبيرة من حركة المرور على الإنترنت. لتلبية طلبات المستخدم لتقديم فيديو عالي الجودة، يلزم إجراء عملية تحسين جودة التجربة (QoE). ولذلك، أجريت أبحاث كثيرة، واستهدفت العديد من النهج المقترحة تعزيز الجودة المتصورة إما بتعزيز استراتيجيات الترميز أو بتحسين آليات التكيف حسب النطاق الترددي المتاح والمهنة العازلة الحالية. ومع ذلك، فإن التغييرات المتكررة في جودة الفيديو لا تزال تمثل تحدياً للعديد من هذه الأساليب ويمكن أن تؤثر على رضا المستخدمين إذا لم يتم التعامل معها.

في هذه الأطروحة، سنصمم طريقة لتحسين جودة تجربة خدمات بث الفيديو المتكيفة لضمان مستوى مناسب من جودة الخدمة للمستخدم النهائي من خلال زيادة جودة استقبال الفيديو والحد من تغييرات الجودة. لبلوغ الهدف المطلوب، سنستخدم نهجاً يعمل على محتويات ذاكرة التخزين المؤقت عن طريق استبدال مقاطع الفيديو المشفرة والتي قد تسبب تغيير الجودة أثناء مشاهدة الفيديو. ويتم تحقيق هذا الحل بمساعدة مكتبة Libdash التي توفر واجهة كائنية التوجه للمعيار الدولي Adaptive Dynamic Streaming عبر (http DASH). وهي خدمة توصيل شائعة الاستخدام تم تطويرها للاستفادة من بث الفيديو بأفضل شكل ممكن من خلال معالجة ظروف عرض النطاق الترددي المختلفة، مما يوفر تدفق سلس.

الكلمات المفتاحية: بث الفيديو التكيفي، بث ديناميكي تكيفي على http، جودة التجربة (QoE)، مخزن مؤقت للوسائط المتعددة.

RÉSUMÉ

Aujourd'hui, le Streaming est devenu un moyen essentiel de diffusion de l'information au public et avec le large déploiement de ses services tels que YouTube, Netflix et autres, il utilise un pourcentage important du trafic Internet. Pour répondre aux exigences des utilisateurs en matière de diffusion de vidéos de haute qualité, un processus d'amélioration de qualité d'expérience est nécessaire. Par conséquent, de nombreuses recherches ont été menées et de nombreuses approches proposées visaient à améliorer la qualité perçue soit en renforçant les stratégies d'encodage, soit en améliorant les mécanismes d'adaptation en fonction de la bande passante disponible et de l'occupation tampon actuelle. Cependant, les changements fréquents de la qualité de la vidéo représentent toujours un défi pour bon nombre de ces approches et peuvent avoir une incidence sur les satisfactions des utilisateurs si elles n'ont pas été traitées.

Dans ce mémoire, en comptant sur une source d'internet illimitée, nous concevons une méthode pour améliorer la qualité de l'expérience des services de streaming vidéo adaptatif afin de garantir un niveau de qualité d'expérience approprié pour l'utilisateur final en maximisant la qualité reçue et en minimisant les changements de qualité. Pour atteindre l'objectif souhaité, nous utilisons une approche qui agit sur le contenu du mémoire tampon en remplaçant les segments vidéo encodés téléchargés qui peuvent provoquer un changement de qualité pendant la vidéo. Cette solution est réalisée à l'aide de la bibliothèque LIBDASH qui fournit une interface orientée objet à la norme internationale Streaming Dynamique Adaptif sur HTTP (DASH). C'est un service de livraison largement utilisé qui a été développé afin de rendre le meilleur streaming vidéo possible en manipulant les conditions de bande passante variables, fournissant ainsi un flux fluide.

Mots clés : Vidéo streaming adaptatif, Dash, la qualité de l'expérience (QoE), Buffer multimédia.

ABSTRACT

Today, Streaming has become an essential means of disseminating information to the public and with the wide deployment of its services such as YouTube, Netflix and others, it uses a significant percentage of Internet traffic. To meet users' requirements for delivering high quality videos, QoE improvement process is required. Therefore, many researches have been conducted and many proposed approaches were aimed to enhance the perceived quality either by strengthening the encoding strategies or by improving the adaptation mechanisms depending on the available bandwidth and the current buffer occupation. However, the frequent changes of video quality still represent a challenge for many of those approaches and it can affect the users' satisfactions if it was not handled.

In this thesis , relying on an unlimited internet source, we design a method to ameliorate the quality of experience of adaptive video streaming services to an appropriate level of quality of experience for the end user by maximize the received quality and minimize quality changes . To reach the desired goal, we use an approach that acts on the buffer content by replacing encoded video chunk that may cause quality change during video .this solution is achieved with the aid of the library LIBDASH that provides an object-oriented interface to the Dynamic Adaptive Streaming over HTTP (DASH) international standard. It is a widely used delivery service that has been developed in order to make the best possible video streaming by handling varying bandwidth conditions, thus providing smooth streaming.

Keywords: Adaptive video streaming, Dash, experience quality (QoE), Multimedia buffer.

TABLE DES MATIÈRES

<i>INTRODUCTION GENERALE</i>	2
CHAPITRE 1. NOTIONS DE BASE	
Introduction.....	5
1.1 Le streaming	5
1.1.1 Définition	5
1.1.2 Les protocoles de transmission	6
1.1.3 Les méthodes de diffusion de Stream	7
1.1.4 L'encodage du contenu	9
1.1.5 Solutions de streaming adaptatif HTTP	10
1.1.6 Avantages du streaming	11
1.2 QoS et QoE pour les services streaming	11
1.2.1 Définitions.....	11
1.2.2 La différence entre la QoS et la QoE	12
1.3 Le standard DASH	13
1.3.1 MPEG-DASH	13
1.3.2 MPD et Segments	15
1.3.2.1 Structure et contenu du MPD	15
1.3.2.2 Les types et les formats de segment	17
1.3.3 Algorithmes d'adaptation au débit binaire.....	19
1.4 Conclusion.....	20
CHAPITRE 2. Etat del'art de vidéo streaming adaptative	
2.1 Introduction	22
2.2 Quelques approches pour l'amélioration du streaming adaptatif :.....	22
2.2.1 Schéma d'adaptation de la qualité vidéo pour améliorer la QoE dans le streaming adaptatif http [25]	22
2.2.2 Adaptation de la qualité à l'aide du codage vidéo évolutif (SVC) en streaming vidéo à la demande (VoD) [26].....	26
2.2.3 MS-Stream : une solution de streaming adaptatif multi-sources améliorant la qualité perçue des consommateurs [4].....	30
2.2.4 Étude comparative	37

2.3	Conclusion.....	40
CHAPITRE 3 : Conception d'un mécanisme d'amélioration de la QoE dans les services de vidéo streaming adaptatif HTTP		
3.1	Introduction.....	42
3.2	La solution.....	42
3.2.1	Principe de la solution.....	42
3.2.2	Description.....	42
3.2.2.1	Adaptation.....	43
3.2.2.2	Révision du flux.....	43
3.2.3	Les paramètres initiaux.....	45
3.2.4	Algorithme de la solution.....	45
3.2.5	L'Organigramme de la solution.....	49
3.3	Conclusion.....	50
CHAPITRE 4. Implémentation et Evaluation		
4.1	Introduction.....	52
4.2	Outils d'implémentation.....	52
4.2.1	La bibliothèque Libdash.....	52
4.2.2	Le lecteur Qtsampleplayer.....	53
4.2.3	Matériel.....	55
4.3	Environnement d'implémentation.....	56
4.3.1	Système d'exploitation.....	56
4.4	Scénario et échantillons du test.....	56
4.4.1	Echantillons utilisés.....	56
4.4.2	Les Scénarios du test.....	57
4.4.2.1	Scenario 1:.....	57
4.4.2.2	Scenario 2:.....	59
4.4.2.3	Scénario 3 :.....	61
4.4.2.4	Scénario 4.....	63
4.4.2.5	Scenario 5.....	67
4.5	Conclusion.....	68
Conclusion générale.....		70
BIBLIOGRAPHIE.....		72

LISTE DES FIGURES

Figure 1. 1 : Streaming média -----	4
Figure 1. 2 : Piles de protocoles de transport vidéo -----	6
Figure 1. 3 : Les standards de codage vidéo. -----	8
Figure 1. 4 : Différence conceptuelle entre QoS et QoE -----	12
Figure 1. 5 : Un scénario de streaming simple entre un serveur HTTP et un client DASH -----	13
Figure 1. 6 : Un système DASH typique -----	14
Figure 1. 7 : Structure d'un MPD général-----	16
Figure 1. 8 : Format du conteneur MP4 -----	17
Figure 1. 9 : Schéma des algorithmes d'adaptation pour DASH -----	18
Figure 2. 1 : Intervalle d'adaptation de la qualité vidéo -----	24
Figure 2. 2 : Algorithm Layer Level Initialization (LLI) -----	27
Figure 2. 3 : Algorithm Layer Level Adjustment (LLA) -----	28
Figure 2. 4 : Présentation de la solution MS-Stream -----	29
Figure 2. 5 : Processus de génération de descriptions multiples à partir du contenu H.264 -----	31
Figure 2. 6 : Exemple de répartition GOP pour créer 3 descriptions différentes -----	31
Figure 2. 7 : Décodage de descriptions multiples avec l'agrégateur de description (DA) -----	31
Figure 3. 1 : Remplacement des segments.....	43
Figure 3. 2 : Organigramme de la solution.....	48
Figure 4. 1 : Architecture MPEG-DASH de haut niveau.....	52
Figure 4. 2 : Configuration de la connexion du client.....	52
Figure 4. 3 : Résultat du test Libdash.....	54
Figure 4. 4 : Interface graphique modifiée du lecteur.....	53
Figure 4. 5 : Résultats de 1 ^{er} scénario	56
Figure 4. 6. Exemple de contenu du tampon avant la révision.....	56
Figure 4. 7 : Exemple de contenu du tampon après la révision.....	56
Figure 4. 8 : Comparaison de la consommation de données.....	58
Figure 4. 9 : Temps nécessaire pour remplacer les segments.....	60
Figure 4. 10 : Résultats de 5 ^{ème} scénario	65

LISTE DES TABLEAUX

Tableau 1.1 : Évolution des normes des codecs de l'UIT-T	10
Tableau 1.2 : Les types de segments.....	18
Tableau 2.1 : Les équations des seuils de commutation vers le haut et vers le bas.....	23
Tableau 2.2 : Les conditions pour une commutation de qualité vidéo.....	24
Tableau 2.3 : Comparaison générale entre les approches.....	37
Tableau 3.1 : Tableau des variables.....	44
Tableau 4.1 : Paramètres des terminaux utilisés.....	54
Tableau 4.2 : Description des scénarios.....	55
Tableau 4.3 : Les vidéos de test.....	56
Tableau 4.4 : Paramètres initiaux du 2 ^{ème} scénario	58
Tableau 4.5 : Résultats du 2 ^{ème} scénario	59
Tableau 4.6 : Résultat du 3 ^{ème} scénario	60
Tableau 4.7 : Résultats du 1 ^{er} cas du 4 ^{ème} scénario.....	62
Tableau 4.8 : Les segments du remplacement téléchargé	64
Tableau 4.9 : Résultats du 2 ^{ème} cas du 4 ^{ème} scénario	64
Tableau 4.10 : Résultats du 3 ^{ème} cas du 4 ^{ème} scénario	65

LISTE DES ACRONYMES

<i>ACRONYMES / ABRÉVIATIONS</i>	<i>LA DESCRIPTION</i>
3GPP	3rd Generation Partnership Project
AAC	Advanced Audio Coding
ACK	Acknowledgement
API	Application Programming Interface
AVC	Advanced Video Coding
CDN	Content Delivery Network
Dash	Dynamic Adaptive Streaming over HTTP
ETSI	European Telecommunications Standards Institute
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
HAS	HTTP Adaptive Streaming
HDS	HTTP Dynamic Streaming
HLS	HTTP Live Streaming
Http	Hypertext Transfer Protocol
Ip	Internet Protocol
Iso	International Organization for Standardization
ITU-T	Telecommunication Standardization Sector of the International Telecommunications Union
MAC	Media Access Control
MEC	Mobile Edge Computing
MPD	Media Presentation description
MPEG	Moving Picture Experts Group
MSS	Microsoft Smooth Streaming
P2P	Peer to Peer

PBT	Bande Passante Totale
PBU	Bande Passante des utilisateurs
QoE	Quality of experience
QoS	Quality of service
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
RSVP	Resource Reservation Protocol.
SNR	Signal-to-Noise Ratio
SVC	Scalable Video Coding
TCP	Transport Control Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locate
VoD	Video On Demand
WebVTT	Web Video Text Tracks
Wifi	Wireless Fidelity
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network
WWAN	Wireless Wide Area Network
XML	Extensible Markup Language

INTRODUCTION GENERALE

Récemment, le trafic mobile a augmenté de manière drastique en raison de la popularité croissante des appareils mobiles, tels que les smartphones et les tablettes PC. Selon les prévisions du trafic mobile de Cisco, le trafic mobile en 2020 est environ six fois plus important qu'en 2016, et le trafic vidéo représente 75 % de ce trafic. Alors la consommation de contenu vidéo devient la méthode préférée. [1]

Au vu de la nature volumineuse des données vidéo, le concept de diffusion en continu (streaming) a rapidement gagné en popularité, car il a permis aux téléspectateurs d'une vidéo de visualiser les données sans avoir à télécharger la vidéo au complet [14]. Les technologies de vidéo streaming envoient le contenu vidéo du serveur vers le client en utilisant le protocole HTTP standard à travers le protocole de transport (TCP)[13], et le lecteur vidéo est soit intégré dans un navigateur Web ou comme une application de streaming.

Avec l'augmentation de la demande pour consommer plus de contenu de meilleure qualité par l'utilisateur, la qualité de l'expérience des utilisateurs (QoE) est devenue un facteur crucial pour le succès des services de streaming vidéo [5]. Pour fournir une qualité d'expérience (QoE) élevée et constante aux utilisateurs un certain nombre de techniques (par exemple, la réservation de ressources, le contrôle d'admission, le streaming adaptatif, etc.) ont été étudiées dans la littérature. Parmi celles-ci, le streaming dynamique adaptatif sur HTTP est une technique de streaming adaptatif basée sur HTTP la plus représentative, et une technique essentielle pour satisfaire la qualité d'expérience des utilisateurs normalisée par le groupe d'experts en images animées (MPEG) [2].

DASH utilise principalement un paradigme basé sur le tirage client pour récupérer des segments vidéo encodés en différents niveaux de débit binaire à partir d'un serveur [3] et les télécharger dans un mémoire tampon en tenant compte de la bande passante ou du tampon disponible. Mais l'un des défis d'un algorithme adaptatif est la variation imprévisible du réseau mobile en termes de bande passante de multiples raisons, ce qui peut entraîner des fluctuations de qualité inutiles [4].

Dans ce contexte, notre projet de fin d'étude consiste à améliorer la qualité de l'expérience des services de streaming vidéo adaptatifs. Nous exploitons la technique de

streaming à débit binaire adaptatif DASH pour maximiser la qualité reçue. Puis nous agissons sur la mémoire tampon des segments vidéo téléchargés afin de minimiser les changements de qualité au cours de la diffusion du média. Le re-téléchargement de segments ne pose pas de problème en termes de consommation de données car nous comptons sur une consommation illimitée.

Notre travail est réparti en quatre chapitres comme suit :

- Le premier chapitre a pour objectif de donner un bref sur les notions de base. Nous sommes intéressés au streaming en premier lieu et nous décrivons certains de ses concepts de base et ses différents avantages, puis nous allons parler sur la qualité d'expérience et la qualité de service en indiquant les différences entre eux .en fin, nous nous terminerons par expliciter le principe du protocole MPEG-DASH.
- Le deuxième chapitre de ce rapport va représenter l'état de l'art. Nous dévoilerons ensuite trois approches connues pour améliorer la qualité d'expérience dans les services de vidéo streaming et les analyser afin de faire une comparaison entre eux.
- Le troisième chapitre va décrire les étapes de conception notre solution « révision du flux vidéo ». nous détaillerons au cours de cette partie le principe de la solution, sa description, les algorithmes utilisés pour l'adaptation de qualité vidéo et pour la révision de la mémoire tampon et en fin nous avons un organigramme pour bien clarifier.
- Le dernier chapitre est le plus important, tous qui concerne l'implémentation et l'évaluation du notre solution est inclus. Nous exposerons alors les différents outils utilisés, l'environnement et les étapes d'implémentation et nous terminons par l'évaluation de la solution proposée à l'aide de scénarios de test qui traitent des différents cas.

Nous terminons par une conclusion générale dans laquelle seront présentées les perspectives à ce travail.

CHAPITRE 1. NOTIONS DE BASE

Introduction

Au cours des dernières années, la majorité des ordinateurs et la quasi-totalité des appareils « mobiles » (tels que les téléphones portables) disposent de moyens de connexion à un ou plusieurs types de réseaux sans fil avec différents standards et ça facilite de bénéficier de plusieurs services.

En conséquence, le nombre d'utilisateurs d'applications de services vidéo sur l'Internet a considérablement augmenté, faisant du streaming vidéo l'un des services multimédias les plus performants.

Dans ce chapitre, Nous allons définir différentes notions de base pour mettre en évidence les caractéristiques du domaine. Premièrement Nous allons introduire des généralités sur les réseaux sans Fils et son classification en examinant à chaque fois les différentes technologies disponibles. Puis une définition de vidéo streaming, ensuite la qualité de service et la qualité d'expérience et on se termine par le standard mpeg-dash.

1.1 Le streaming

1.1.1 Définition

Le Streaming consiste à découper les données en paquets dont la taille est adaptée à la bande passante disponible entre le client et le serveur. Quand le client a reçu suffisamment de paquets (buffering), l'application cliente commence à jouer un paquet, décompresse un autre et reçoit un troisième. Ainsi la technologie du streaming permet de commencer à visualiser un contenu multimédia sans être obligé de télécharger la totalité des données (on fonctionne en flux continu). [5]

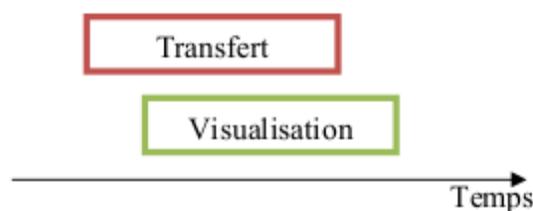


Figure 1. 1. Streaming média [6]

Le streaming peut être divisé en deux grandes catégories : à la demande et en temps réel. Dans le cas du streaming à la demande, le fichier vidéo est stocké sur un serveur et les clients le demandent (YouTube par exemple). Dans le cas du streaming en temps réel, le contenu est généré et transmis à la volée avec un retard léger et constant (par exemple un évènement de football en direct). Le streaming en temps réel peut être point à point ou diffusé. Tant à la demande qu'en temps réel, un tampon est géré pour stocker les données reçues avant qu'elles ne soient affichées [6].

Le streaming implique des protocoles à différentes couches telles que la couche transport, session, présentation et application. Une technologie de streaming vidéo peut être définie par les protocoles utilisés et la façon dont il diffuse les médias.[7]

1.1.2 Les protocoles de transmission

Il y'a le protocole TCP et le protocole UDP dans la couche transport. TCP est un protocole basé sur un flux, il fournit les algorithmes efficaces de contrôle d'erreur et de contrôle de congestion, les utilisateurs n'ont pas besoin de prendre en compte la perte de paquets et l'ordre d'arrivée des données. Le protocole UDP est basé sur la transmission par blocs, il ne garantit pas la perte de données, l'erreur et l'ordre d'arrivée des données, en même temps, il n'a pas d'algorithme de contrôle de congestion. Le protocole UDP est plus adapté à une utilisation dans le système de diffusion de médias en continu. Le protocole UDP présente les avantages suivants [8] :

1. L'UDP est orienté vers le sans connexion, il n'y a pas de processus de handshake TCP à trois voies, la vitesse est rapide et le coût de la maintenance de l'état des sessions UDP est plus faible.
2. L'erreur multimédia n'affecte que la qualité du visionnage, il n'est pas nécessaire de retransmettre après que l'erreur se soit produite, et la retransmission des erreurs TCP n'est pas contrôlée.
3. En raison de l'instabilité du réseau, il est facile de produire de grandes fluctuations du débit, qui est déterminé par l'algorithme de contrôle de la congestion du TCP. Lorsque les paquets ne répondent pas pendant une longue période, le TCP réduit le débit, puis le délai de transmission augmente.

«Selon le protocole de communication TCP, une connexion entre deux hôtes s'établit en trois étapes : c'est le **three-way handshake**. »

4. UDP peut effectuer une traversée NAT, tandis que TCP est plus difficile.
5. UDP est plus facile à contrôler la taille du contrat de sous-traitance que TCP, car TCP est un protocole de communication orienté flux, il n'y a pas de concept de paquets à la couche application, et la taille du contrat de sous-traitance du bas est contrôlée par le système d'exploitation.

Cependant, étant donné que les communications basées sur l'UDP sont souvent interceptées par des pare-feu et/ou des NAT, il existe des environnements lorsque les services de streaming vidéo UDP ne peuvent pas être offerts. TCP, d'autre part, peut facilement contourner les pare-feu et les NAT. Pour cette raison, la plupart des services de streaming vidéo actuels, tels que YouTube et Niconico, support basé sur TCP streaming [22]. Donc, afin de faciliter la traversée des NAT et des pare-feu, HyperText Transfer le protocole (HTTP) a généralement été utilisé sur TCP [9].

UDP est aussi utilisée en conjonction avec des protocoles de couche d'application spécifiques, appelés protocole de transport en temps réel RTP [10].

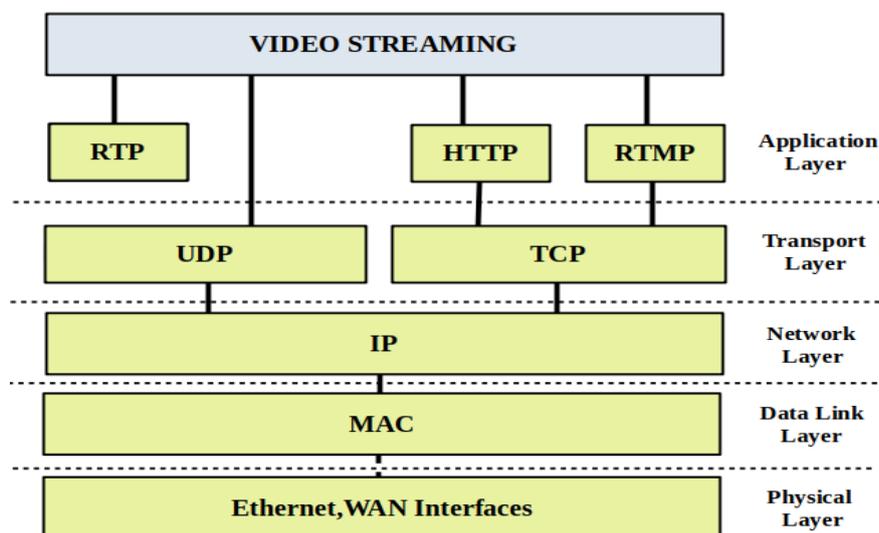


Figure 1.2 .Piles de protocoles de transport vidéo [9]

Network address translation (NAT) ça fonction dans un routeur de service intégré traduit une adresse IP source interne en adresse IP globale.

a. **Téléchargement progressif**

Le téléchargement progressif n'utilise pas de protocoles spécifiques. Il suit le concept de streaming, ce qui signifie que les utilisateurs peuvent commencer à regarder la vidéo avant qu'elle ne soit complètement téléchargée. Le service permet aux utilisateurs de télécharger le fichier vidéo le plus rapidement possible en fonction de la largeur de bande passante du réseau. Un délai initial est introduit au début de la session vidéo afin d'accumuler une certaine quantité de données vidéo qui empêche la famine de la mémoire tampon (mémoire tampon vide).

La famine tampon peut cependant se produire parce que la qualité de la vidéo n'est pas adaptée à la largeur de bande passante du réseau.[11]

b. **Real-time Transport Protocol (RTP) :**

Le protocole de transport en temps réel (RTP) est un protocole de transport basé sur l'UDP conçu spécifiquement pour les transferts en temps réel [1]. Le RTP est associé au protocole de contrôle en temps réel (RTCP) pour contrôler la qualité du service et pour transmettre des informations sur les participants dans une session en cours. Il fournit un retour d'information sur la qualité des données transmises, permettant ainsi des actions telles que l'ajustement du débit de données. [6]

Le protocole RTSP (Real Time Streaming Protocol) est un protocole de couche de présentation utilisé avec le RTP. Il est similaire au protocole HTTP (Hyper Text Transfer Protocol), il transmet les demandes des clients et lance des activités telles que la lecture, la pause et l'enregistrement. Le protocole de réservation des ressources (RSVP) est utilisé pour établir les sessions. RTP est utilisé par Flash, RealPlayer, Windows Media Player et QuickTime player.[6]

c. **HTTP Adaptive Streaming (HAS) :**

Le streaming adaptatif HTTP (HAS) utilise le protocole HTTP sur TCP. C'est le nouveau mode de diffusion de contenu vidéo qui a suscité l'intérêt du marché ces dernières années. Il utilise le protocole HTTP pour contourner efficacement les NAT et les pare-feu. De plus, le retard de retransmission des paquets TCP peut être absorbé par le déploiement d'un tampon côté client. En divisant le clip vidéo en petits segments et en fournissant plusieurs niveaux de qualité par segment, le client peut adapter dynamiquement la qualité aux conditions actuelles du réseau. La technologie

de streaming adaptatif HTTP est assez simple et présente l'avantage de réutiliser les réseaux de distribution de contenu (CDN), les proxys et les caches existants et rentables basés sur HTTP. [12]

1.1.3 L'encodage du contenu

La transmission de contenus vidéo non compressés nécessite une quantité énorme de ressources. Par exemple, avec une résolution de 640x480 et une taille d'image de 300Kb (basse qualité), une minute de vidéo est supérieure à 1Gbytes. Par conséquent, nous devons réduire la taille de la vidéo avant la transmission par le réseau, ce processus est appelé vidéo codage ou compression vidéo. Il y a le processus de codage et de décodage. Le logiciel/matériel responsable du processus d'encodage est le codec. L'efficacité de la transmission vidéo dépend fortement de l'efficacité du codec vidéo [6]. Nous examinons les codecs vidéo dans la section suivante.

- **Codec vidéo :**

Au niveau international, les deux organismes les plus actifs pour la normalisation des systèmes de compression vidéo sont VCEG (Video Compression Expert Group) de l'ITU-T (International Telecommunication Union-Telecommunication) et MPEG (MovingPictures Expert Group) de l'ISO/IEC (International Organization of Standardization /International Electro-Technical Commission) comme il est expliqué dans la figure suivante. [13]

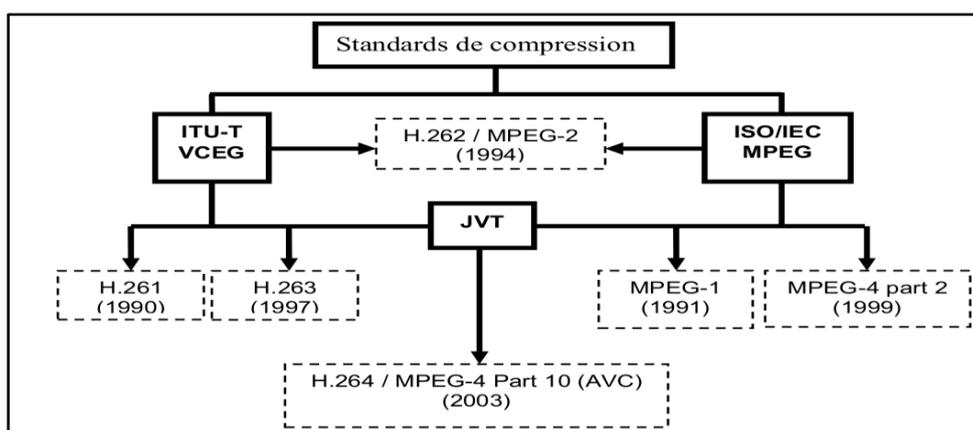


Figure 1. 3. Les standards de codage vidéo.

Une liste non exhaustive des codecs vidéo les plus utilisés actuellement est comme le tableau suivant montre :

Nom	Date	Principales caractéristiques
H261	1990	ISDN vidéo conférence
H263	1996	Basé sur H261 avec une performance de codage accrue
MPEG-1	1991	Supports de stockage numériques, CD-ROM, au débit binaire jusqu'à 1,5
MPEG-2(H262)	1994	Des débits de données plus élevés pour HDTV, SDTV, DVD
MPEG-4 Part 2	2000	Application mobile et streaming
H264 (MPEG-4 Part 10 codage vidéo avancé)	2005	Radiodiffusion, télévision, DVD haute définition, stockage numérique, application mobile

Tableau 1. 1. Évolution des normes des codecs de l'UIT-T [6]

Donc on peut penser que la diffusion vidéo en continu se compose des étapes suivantes [14] :

- 1) Positionner la vidéo compressée en paquets
- 2) Commencer la livraison de ces paquets
- 3) Commencez le décodage et la lecture au niveau du récepteur alors que la vidéo est immobile en cours de livraison

1.1.4 Solutions de streaming adaptatif HTTP

Au cours des dernières années, plusieurs solutions de streaming adaptatif ont été proposées par différents organismes. Ces techniques s'améliorent au fil du temps [15].

- **Solutions propriétaires** : [15]

HLS (HTTP Live Streaming) by Apple.

MSS (Microsoft Smooth Streaming) by Microsoft.

HDS (HTTP Dynamic Streaming) by Adobe.

- **Solution standard internationale** : [15]

DASH (Dynamic Adaptive Streaming over HTTP) by MPEG and ISO/IEC

1.1.5 Avantages du streaming

La transmission de données multimédias en streaming présente plusieurs avantages [17] :

1. Il permet aux usagers de commencer à visualiser, ou entendre un contenu audiovisuel, volumineux par nature, sans attendre que la totalité des données soit téléchargée.
2. Il offre la possibilité au spectateur de se rétracter à tout moment, si le contenu ne correspondait pas à ce qu'il attendait.
3. Il ne nécessite pas de disposer d'un espace de stockage chez l'utilisateur. Toutefois, un espace limité de mémoire tampon peut être demandé pour le décodage des données.
4. Aucune copie locale n'est conservée. Ceci est très utile pour le partage de contenus soumis à des licences restrictives quant à la distribution de contenus (mais qui en autorise la diffusion).
5. Il est bien adapté pour la transmission des événements en direct.

L'évaluation de l'expérience des individus est un point de grande importance lorsqu'ils interagissent avec les différentes applications dans le contexte de streaming adaptatifs, et pour donner la satisfaction à l'utilisateur final la notion de QoE est utilisée.

1.2 QoS et QoE pour les services streaming

Jusqu'à récemment, la qualité de service (qos) fournie était évaluée d'un point de vue technique pour déterminer les performances du réseau. Néanmoins, pour de nombreux services tels que le streaming vidéo, la qualité de service ne peut refléter l'influence de la fluctuation du réseau sur l'expérience de l'utilisateur. [17]

Donc la satisfaction des utilisateurs n'est plus prise en compte. Étant donné que de nombreux fournisseurs de services se font concurrence pour un plus grand nombre de clients, une nouvelle notion est apparue : la qualité d'expérience (qoe), utilisée à la place de la qualité de service pour améliorer le service et obtenir le retour d'information du consommateur sur un service spécifique (par exemple, le réseau) [31].

1.2.1 Définitions

➤ **La qualité du service :**

La qos fait référence à la capacité d'un réseau à obtenir un comportement plus déterministe, de sorte que les données peuvent être transportées en optimisant des paramètres tels que la perte de paquets, le retard, la gigue et la consommation de bande passante. Il convient de noter que la qualité de service ne prend pas en compte la perception de l'utilisateur final [37]. Donc, simplement qos est la capacité d'un réseau de fournir un meilleur service au trafic réseau sélectionné.[30]

➤ **La qualité d'expérience QoE :**

La qoe est un nouveau sujet de recherche qui est actuellement traité par les différents forums de normalisation. Elle décrit le degré de satisfaction de l'utilisateur d'un service, influencé par le contenu, le réseau, l'appareil, l'application, les attentes de l'utilisateur, et les objectifs, et le contexte d'utilisation. [17]

La qoe est au cœur de la conception, de la création, de l'approvisionnement et de la gestion des services multimédias actuels et futurs, en particulier lorsqu'ils sont fournis sur Internet, ce qui est le cas le plus courant. [30]

En effet, la qoe est un concept complexe qui se situe à la jonction de plusieurs disciplines scientifiques, techniques et humaines, pour la plupart non liées. [17]

La qoe pour les applications multimédias de prochaine génération sera une combinaison de paramètres de qos mesurables prenant en compte à la fois l'environnement de réseau et de service et les paramètres de non-qualité. [30]

1.2.2 La différence entre la QoS et la QoE

La qualité du service est exprimée par les paramètres du réseau comme la bande passante, le délai, la gigue ou la perte de paquets [6]. Du point de streaming vidéo, QoS a indiqué à la capacité du réseau à transmettre des paquets de la meilleure façon possible. Donc les mesures de la QoS sont basées sur l'évaluation expérimentale, technologique ou simulée. [18]

Cependant, un bon QoS ne garantit que tous les clients aient l'expérience du service pour être bon. Attendu que QoS représente une vision centrée sur le réseau des performances du système, la qualité de l'expérience représente la vue centrée sur l'utilisateur (Fig.1.6). Par conséquent, la qualité de l'expérience exprime directement la satisfaction de l'utilisateur d'un service. Il est influencé par le système, les utilisateurs et le contexte [38]. Ses mesures sont basées sur une approche méthodologique et multidisciplinaire. [18]

Au niveau des applications multimédias, le temps nécessaire à un service de vidéo à la demande pour démarrer la diffusion, le nombre d'événements de blocage rencontrés lors d'un service de streaming vidéo, la résolution des images affichées sur un site de réseau social, la façon dont différentes images sont combinées dans un portail d'actualités et le type de contenu qu'ils véhiculent, ainsi que de nombreux autres facteurs du système ont tous un impact sur la qualité perçue. [30]

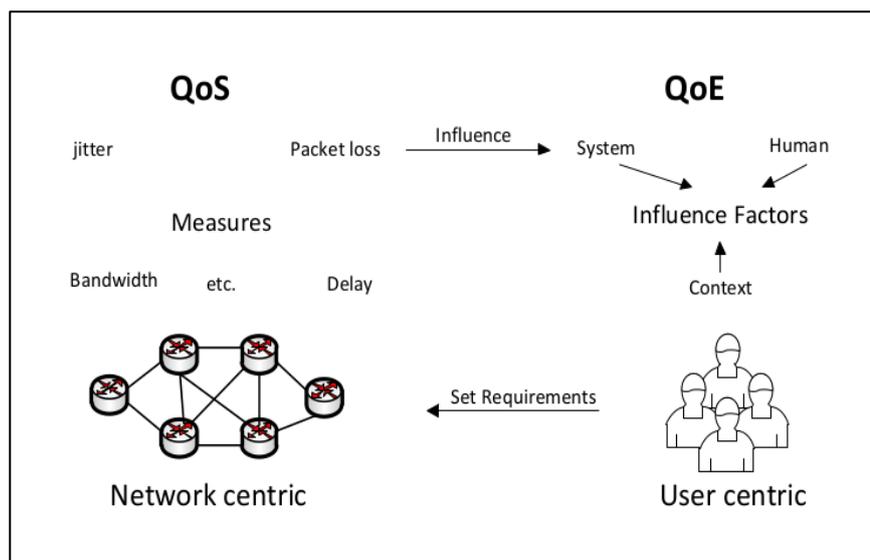


Figure 1. 4. Différence conceptuelle entre QoS et QoE [38].

1.3 Le standard DASH

1.3.1 MPEG-DASH

Le streaming HTTP en direct d'Apple, le streaming fluide de Microsoft et le streaming dynamique HTTP d'Adobe utilisent tous le streaming HTTP comme méthode de livraison.

Mais chacun des protocoles de streaming a ses propres formats de manifeste et de segment. Ainsi, chaque fois que le contenu reçu du serveur, l'appareil doit prendre en charge le protocole client, par exemple HLS ne sera pris en charge que par les appareils Apple, il ne peut pas être lu sur d'autres appareils Android. Il n'y a pas d'interopérabilité entre les appareils et les serveurs de différents fournisseurs. MPEG-DASH est donc celui qui assure l'interopérabilité entre divers serveurs et périphériques clients. [31]

Le travail de bout en bout de l'ensemble du système MPEG-DASH comprend différentes parties, y compris des composants de génération de contenu, d'hébergement de contenu, de livraison de contenu et de consommation de contenu [16].

La norme [ISO23009] ne définit pas tous les composants du système, elle se concentre principalement sur les formats des fichiers liés MPEG-DASH. Ils sont répertoriés dans un rectangle en pointillés sur la figure 1.7, l'un est le fichier MPD (Media Description Presentation) et l'autre est les segments. [1]

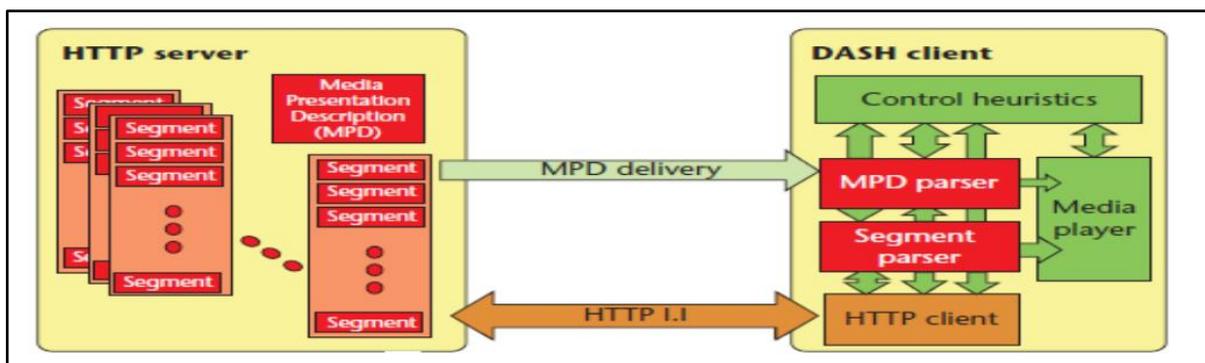


Figure 1. 5. Un scénario de streaming simple entre un serveur HTTP et un client DASH [31].

Dans cette figure, le contenu multimédia est capturé et stocké sur un serveur HTTP et livré via HTTP. Le contenu existe sur le serveur en deux parties : Media Presentation Description (MPD), qui décrit un manifeste du contenu disponible, ses différentes alternatives, leurs adresses URL et d'autres caractéristiques ; et des segments, qui contiennent les flux binaires multimédias réels sous forme de blocs, dans des fichiers simples ou multiples. [31]

L'adoption d'une solution HTTP basée sur le streaming adaptatif est motivée par le fait que le streaming non adaptatif subit des retards de démarrage et une mise en mémoire tampon

importants pendant la session vidéo. Cela est dû à la variation de la capacité du réseau au fil du temps. [6]

Les autres cibles clés de DASH sont : l'insertion simple de publicités et l'utilisation de CDN, proxy et caches HTTP existants et économiques. [6]

Dans DASH, le fichier multimédia est divisé en petits segments de durée fixe, par exemple 2 secondes, qui peuvent être encodés à différents débits ou qualités (appelées représentations). Les segments sont fournis sur un serveur web et peuvent être téléchargés via des requêtes GET conformes aux normes HTTP (Figure 1.8). L'adaptation au débit binaire est effectuée du côté client pour chaque segment, par exemple, le client peut passer à un débit binaire plus élevé - si la bande passante le permet - sur une base par segment (Figure 1.11). Les relations temporelles et structurelles entre les segments sont décrites dans le fichier MPD. [38]

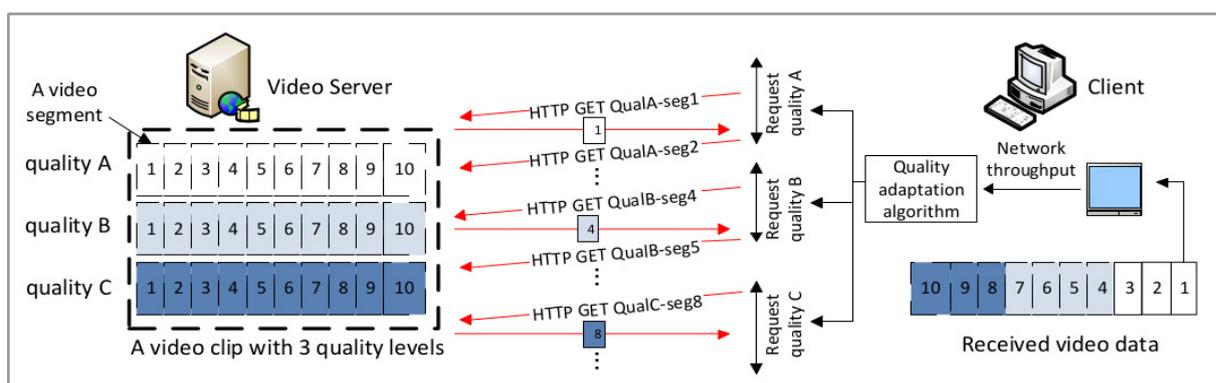


Figure 1. 6. Un système DASH typique [4]

1.3.2 MPD et Segments

MPEG-DASH définit le fichier XML Media Presentation Description et les formats binaires des segments. Le format binaire de segments se réfère au format conteneur vidéo, MPEG-DASH ne définit pas les codecs pour la vidéo et l'audio à l'intérieur du format conteneur. MPEG-DASH est agnostique aux codecs vidéo et audio, et de nombreux autres codecs peuvent être utilisés. [16]

Une description de présentation multimédia (**MPD**) décrit les informations de segment (synchronisation, URL.....).

1.3.2.1 Structure et contenu du MPD

MPEG-DASH utilise le fichier manifeste MPD dans lequel la structure du contenu audiovisuel est décrite et fournit au client l'emplacement où les fichiers spécifiés sont stockés. Il existe deux types de MPD : [15]

- **MPD statique** : il ne change pas avec le temps, utilisé pour la vidéo à la demande, mais il peut également être utilisé pour du contenu en direct. Il n'est envoyé au client qu'une seule fois au début de la session.
- **MPD dynamique** : lorsqu'un nouveau contenu est disponible, le MPD est mis à jour. Par conséquent, le client doit périodiquement demander le MPD car de nouveaux contenus peuvent être disponibles, utilisé pour le contenu en direct.

La structure du MPD est la suivante : [15]

- **Périodes** : décrire une partie du contenu avec l'heure de début et la durée de ce contenu. Plusieurs périodes peuvent être utilisées pour diviser le contenu multimédia en différentes scènes ou chapitres et pour séparer le contenu principal de la publicité ou d'autres contenus non liés au contenu principal.
- **Ensemble d'adaptation** : contient un ensemble de flux qui appartiennent au même service.
Dans le cas le plus simple, une période peut contenir deux ensembles d'adaptation, un pour la vidéo et un pour l'audio. S'il y avait plusieurs audio (par exemple, avec différentes langues ou différentes descriptions audio), chacun serait un ensemble d'adaptation différent. Un ensemble d'adaptation peut contenir d'autres types de données, tels que des sous-titres ou d'autres types de contenu liés à l'accessibilité. Le client utilise ses préférences pour choisir les ensembles d'adaptation qui répondent le mieux à ses besoins (langues ou services d'accessibilité).
- **Représentations** : les représentations permettent à un ensemble d'adaptation d'avoir le même contenu encodé avec différents paramètres. Par exemple, dans le cas de la vidéo, différentes résolutions et différents débits binaires sont généralement utilisés. Les représentations peuvent également être codées à l'aide de différents codecs, ce qui permet la prise en charge des clients avec différents codecs compatibles.

- **Segments** : chaque représentation est formée d'un ou plusieurs segments. Chaque Segment représente chacune des parties dans lesquelles le contenu multimédia a été divisé.

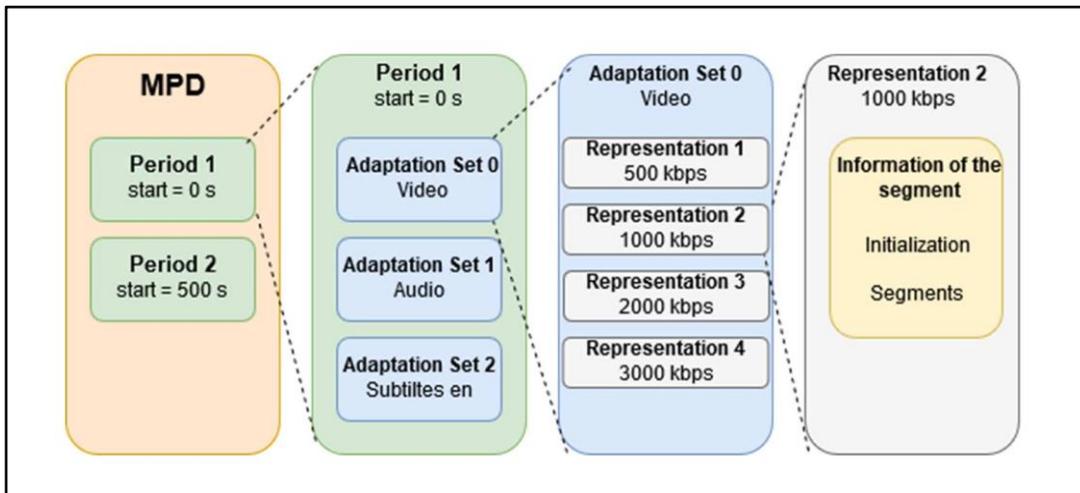


Figure 1. 7. Structure d'un MPD

La norme DASH identifie aujourd'hui deux profils possibles [20] :

- (1) **Un profil VOD** qui nécessite de manipuler coté serveur une table d'indexation ainsi que des fichiers (un fichier par représentation), selon le niveau de qualité requis par le client (par exemple), la table d'indexation sélectionne un segment identifié par un octet de début et un octet de fin à l'intérieur du fichier correspondant à la représentation demandée.
- (2) **Un profil Live** qui permet de manipuler directement des segments au sens défini dans ce profil autorise deux types d'implémentations, l'un avec une table d'indexation qui identifie des numéros de segments, l'autre par tronçon de temps (segment time line), dans ce second mode le premier segment correspond par exemple aux dix premières secondes de la vidéo, le second segment aux dix suivantes, etc.

1.3.2.2 Les types et les formats de segment

La partie formats de segment de la norme MPEG-DASH définit principalement les types de segment et les formats utilisés dans ces segments. Les types de segments sont résumés et énumérés dans le tableau suivant [16] :

Type	Description
Segments d'initialisation	contient des informations pour l'initialisation, ne doit contenir aucune donnée de médias
Segments médias	Contient des flux de médias réels
Segments d'indice	contient des informations d'indexation pour les segments de média
Segments de commutation BitStream	contient des informations permettant de passer à la représentation assignée

Tableau 1. 2. Les types de segments

➤ **Format du contenant :**

Le fichier conteneur est utilisé pour identifier et entrelacer différents types de données, par exemple dans un fichier vidéo, il peut contenir des images, audio et métadonnées. Le conteneur ne décrit pas comment les données à l'intérieur sont encodées, donc un programme peut encore avoir besoin de codec spécifique pour décoder les données contenues, même s'il est capable d'identifier et d'ouvrir un fichier conteneur. MPEG-DASH crée la norme et rend le vidéo agnostique codec [31].

Dans la norme, deux formats de conteneur MPEG sont énumérés, ISO Base Media File Format et MP4 (MPEG-4 Partie 14). Pour une paire de format conteneur vidéo et codec vidéo, MP4 est le plus largement utilisé et il est basé sur le ISO Base Media File Format [31].

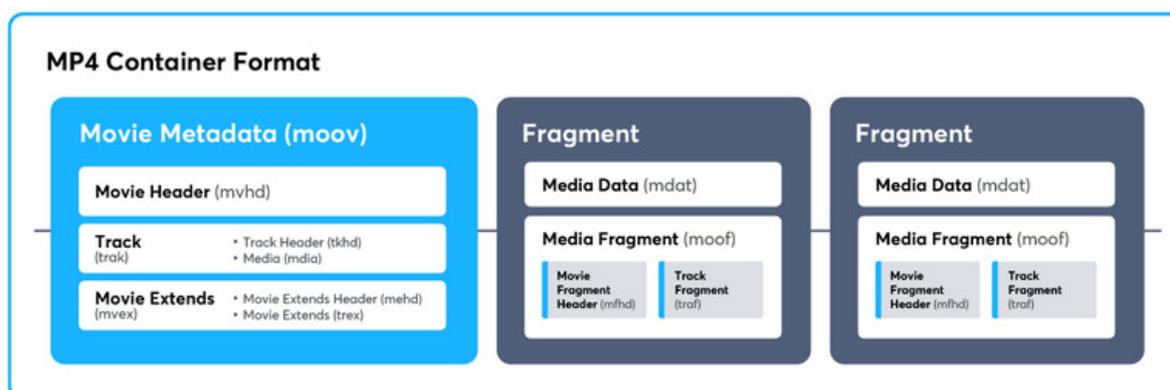


Figure 1. 8. Format du conteneur MP4 [24]

1.3.3 Algorithmes d'adaptation au débit binaire

DASH ne dicte pas la logique d'adaptation, cette complexité est déplacée du côté client. L'avantage est que la logique de streaming basée sur le client permet une grande évolutivité et flexibilité. Par conséquent, les algorithmes d'adaptation au débit binaire ne sont pas normalisés dans DASH, le but étant de choisir un débit binaire assurant une bonne qualité vidéo et prévenant les interruptions inutiles de lecture vidéo. [6]

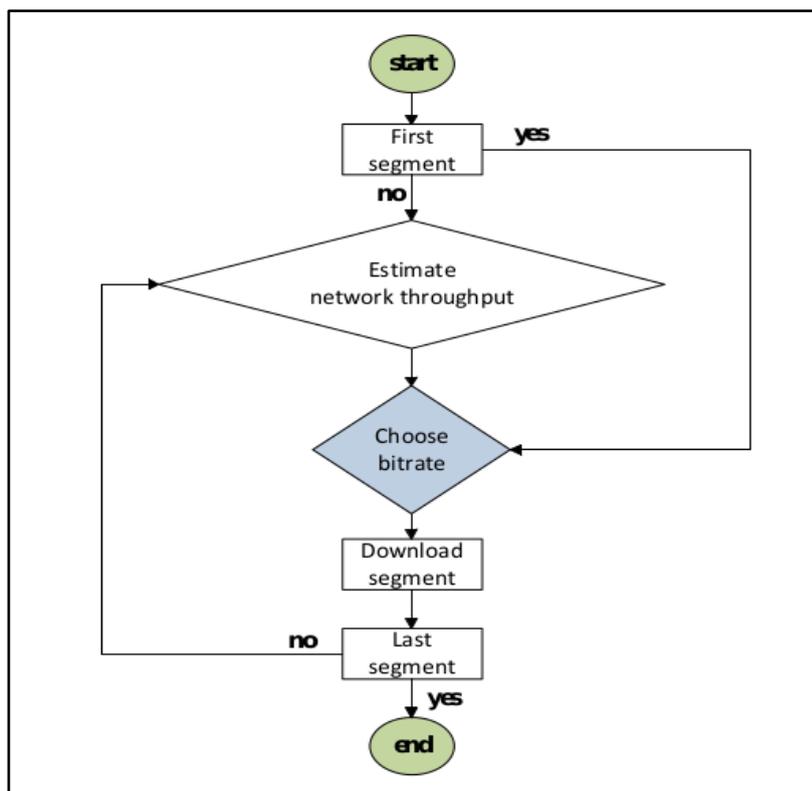


Figure 1. 9. Schéma des algorithmes d'adaptation pour DASH [6]

Les méthodes d'adaptation de la qualité se divisent habituellement en deux catégories : les algorithmes basés sur le débit (TBA) et les algorithmes basés sur le tampon (BBA) [6].

(1) Adaptation basée sur le débit (TBA) :

TBA lance la transmission en demandant le débit binaire disponible le plus bas et, sur la base du débit réseau moyen, les segments suivants seront sélectionnés de manière additive croissante et multiplicative décroissante (AIMD). [21]

(2) Adaptation basés sur les tampons (BBA) :

L'adaptation basée sur les tampons vise à maintenir l'occupation des tampons à un niveau souhaité. Il s'agit généralement d'une méthode de tampon fixée puisque les seuils de tampon sont définis et que le débit binaire change en fonction du niveau du tampon. Ainsi, BBA utilise uniquement l'occupation tampon pour demander le segment initial et ensuite, si nécessaire, estime la capacité. [21]

1.4 Conclusion

Dans ce chapitre, nous avons dans un premier temps donné des détails sur le streaming vidéo, ses protocoles de transmission ainsi les méthodes de diffusion, puis qos et qoe pour les services de streaming et la différence entre eux et on se termine par le standard DASH.

CHAPITRE 2. Etat de l'art de vidéo streaming adaptative

2.1 Introduction

Comme nous l'avons mentionné précédemment, QoS a été utilisé pour désigner l'amélioration de la performance réalisée par le matériel et / ou des logiciels mais ses métriques permettent de garantir les services ne parviennent pas à mesurer la subjectivité associée à la perception humaine et pour muter vers des politiques fondées sur une approche globale de la qualité de bout-en-bout, la QoE est apparue.

Avec l'hétérogénéité des réseaux et la demande croissante de services multimédias, la qualité d'expérience (QoE) devient un facteur déterminant de réussite ou d'échec de ces services [28]. Plusieurs approches ont été proposées pour l'améliorer. La plupart de ces approches ont apportent des changements et des améliorations soit sur l'adaptation du débit binaires ou sur les stratégies d'encodage.

Dans ce chapitre on va étudier trois approches connues pour améliorer la qualité d'expérience dans les services de vidéo streaming ainsi les avantages et les inconvénients de chaque approche, et on se termine par une étude comparative entre eux.

2.2 Quelques approches pour l'amélioration du streaming adaptatif :

2.2.1 Schéma d'adaptation de la qualité vidéo pour améliorer la QoE dans le streaming adaptatif http [25]

Les schémas d'adaptation de la qualité vidéo conventionnels basés sur la mesure de la bande passante sont trop agressifs pour répondre à la variation à court terme de la bande passante du réseau, des changements fréquents de la qualité vidéo se produisent. En outre, ils ne reflètent pas l'occupation du tampon et les discontinuités de lecture se produisent en raison du sous-débit du tampon. En conséquence, ces schémas conventionnels peuvent difficilement garantir la QoE du streaming adaptatif HTTP.

Dans cet article, ils proposent un schéma d'adaptation de la qualité vidéo côté client pour améliorer la qualité d'expérience de la diffusion en continu adaptative HTTP en tenant compte à la fois de la variation de la largeur de bande du réseau et de l'occupation de la mémoire tampon du client.

1. Formulation de la solution

Le schéma proposé calcule le débit du segment pour mesurer la bande passante du réseau et détermine le seuil d'occupation du tampon. Ce seuil reflète le débit binaire de la qualité vidéo. Enfin, leur schéma adapte la qualité vidéo en fonction de la mesure de la bande passante et de l'occupation du tampon du client. Ainsi, ce schéma peut être appliqué en 3 phases :

a. Mesure de la bande passante :

Traditionnellement, la plupart des schémas d'adaptation de la qualité vidéo du streaming adaptatif HTTP mesurent le débit des segments désigné par « $T_{hseg} [i]$ » pour un streaming vidéo sans faille chaque fois qu'un segment est arrivé au client.

b. Déterminer le seuil d'occupation du tampon :

En principe, la qualité vidéo augmente en raison de l'augmentation transitoire de la bande passante du réseau, ce qui provoque des changements inutiles de la qualité vidéo. Cependant, le schéma proposé n'augmente pas la qualité vidéo lorsque l'occupation du tampon est inférieure au seuil, même si la bande passante du réseau est augmentée. Nous déterminons le seuil d'augmentation de la qualité vidéo. Ce seuil est déterminé en tenant compte de la qualité vidéo supérieure suivante, qui est calculée comme l'équation (1) montré dans le tableau2.1

La diminution de la qualité vidéo due à la réduction transitoire de la bande passante du réseau entraîne des changements inutiles de la qualité vidéo. Cependant, le schéma proposé ne diminue pas la qualité vidéo lorsque l'occupation du tampon est supérieure au seuil, même si la bande passante du réseau est diminuée. Le seuil de diminution de la qualité vidéo indiquée par B est donné par l'équation (2) montré dans le tableau2.1 :

Numéro	L'équation
(1)	$B_{up} = \frac{\sum_{k=1}^{cur+1} R_k}{\sum_{K=1}^L R_k} \cdot B_{max}$
(2)	$B_{down} = \frac{\sum_{k=1}^{cur} R_k}{\sum_{K=1}^L R_k} \cdot B_{max}$

Tableau 2. 1. les équations des seuils de commutation vers le haut et vers le bas

B_{up} : seuil d'augmentation de la qualité vidéo .

R_k : débit binaire du $k^{ème}$ niveau de qualité vidéo.

B_{max} :seuil maximal d'occupation du tampon.

L : le niveau de qualité vidéo maximum.

c. Adaptation de la qualité vidéo :

Le schéma proposé déploie deux mécanismes pour adapter la qualité vidéo. Le premier mécanisme est commutateur progressif de haut en bas ou de bas vers haut. Le deuxième mécanisme est agressif vers le bas. Passer vers le haut consiste à sélectionner la prochaine qualité vidéo supérieure, et vers le bas consiste à sélectionner la prochaine qualité vidéo inférieure. Une baisse agressive consiste à sélectionner le débit binaire minimum de la qualité vidéo. Ces mécanismes ont l'avantage que le schéma proposé peut augmenter la qualité vidéo en douceur tout en maintenant une occupation de tampon suffisante. En outre, le schéma proposé peut diminuer rapidement la qualité vidéo lorsque la bande passante du réseau diminue considérablement. La commutation vers le haut se produit lorsque les conditions (1) et (2) montrés dans le tableau2.2 sont remplies. En revanche, la commutation vers le bas se produit lorsque les conditions (3) et (4) sont remplies.

Si le schéma proposé diminue la qualité vidéo d'un niveau lorsque la bande passante du réseau diminue fortement, il peut provoquer un dépassement de mémoire tampon. Afin d'éviter un débordement de tampon, notre système procède à une baisse agressive. Une baisse agressive se produit lorsque la condition (5) de tableau2.2 est remplie.

Numéro	Condition
(1)	$TH_{seg}[i] > R_{cur+1}$
(2)	$B_{cur}[i] > B_{up}$
(3)	$TH_{seg}[i] < R_{cur}$
(4)	$B_{cur}[i] < B_{down}$

(5)	$B_{cur}[i-1] - B_{cur}[i] > \frac{R_{cur}}{R_{min}} \cdot MSD$
-----	---

Tableau 2. 2. Les conditions pour une commutation de qualité vidéo

R_{min} : Le débit binaire minimum de la qualité vidéo

R_{cur+1} : Le débit binaire supérieur suivant de la qualité vidéo.

$B[i-1]$: L'occupation du tampon lorsque le (i-1)^{ème} segment est téléchargé.

$B_{cur}[i]$: Occupation de la mémoire tampon lors du téléchargement du i^{ème} segment.

MSD : Durée du segment média.

Le schéma proposé maintient la qualité vidéo pendant que le client reçoit N segments. Donc il n'adapte pas la qualité vidéo tant que le client n'a pas reçu N segments. N est calculé comme suit :

$$N = \left\lfloor \frac{R_{min}}{R_{cur}} \cdot \frac{1}{MSD} \cdot B_{cur} \right\rfloor$$

Et cela est illustré dans la figure 2.1 suivante :

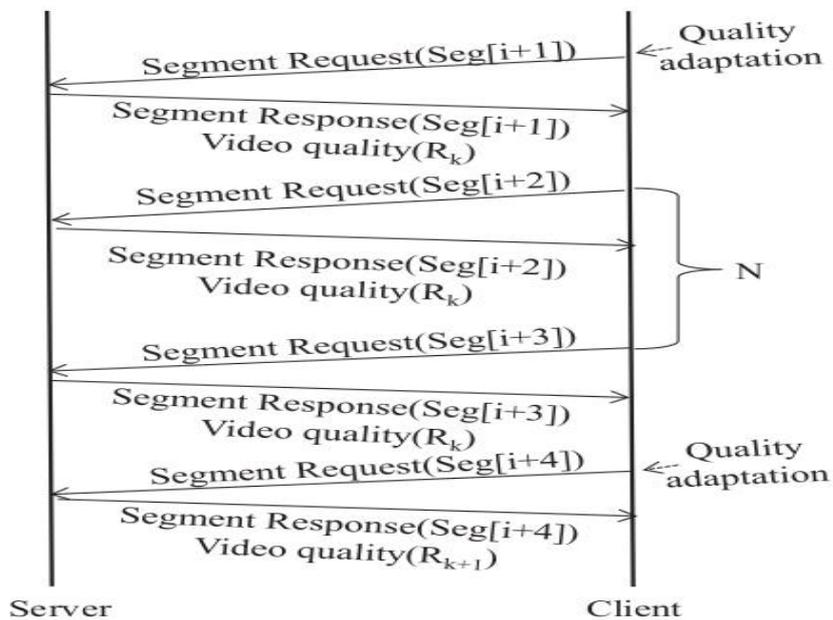


Figure 2. 1. Intervalle d'adaptation de la qualité vidéo

Avec : $Seg[i+1]$: le (i+1)^{ème} segment.

2. Les avantages de l'approche :

- comme tous les approches basée sur HTTP, cet approche proposée peut traverser facilement les pare-feu et réutiliser l'infrastructure HTTP déjà déployée comme les serveurs HTTP, les proxys HTTP et les nœuds CDN (Content Delivery Network).
- Décision d'adaptation basée sur l'observation locale.
- Un client récupère chaque bloc indépendamment et maintient l'état de la session de lecture tandis que les serveurs n'ont pas besoin de ne conserver aucun état.
- Aucun composant côté serveur spécial n'est requis.
- La solution est implémentée au niveau de la couche application.

3. les inconvénients de l'approche :

- Si le serveur critique tombe en panne, les demandes des clients ne sont pas satisfaites.
- Les débits vidéo supérieurs à la bande passante disponible ne sont jamais autorisés à être sélectionnés pour éviter les interruptions de lecture.
- L'estimation de la bande passante dans la couche application est assez imprécise en raison de sa limitation architecturale.
- l'avis du l'utilisateur n'est pas pris en compte.

2.2.2 Adaptation de la qualité à l'aide du codage vidéo évolutif (SVC) en streaming vidéo à la demande (VoD) [26]

L'idée de base de cette architecture proposée est d'utiliser efficacement les ressources disponibles de chaque homologue afin de maximiser la qualité du flux délivré sous des variations de ressources. Tirant parti de l'évolutivité de la conception SVC, la diffusion P2P adaptative de qualité pourrait décider quelle couche vidéo correspond le mieux aux ressources disponibles des pairs, qui est responsable de prendre une décision sur la meilleure correspondance des couches spatiales, temporelles et de qualité.

Nouveaux mécanismes d'adaptation de la qualité utilisant le codage vidéo évolutif (SVC) sur des systèmes de vidéo à la demande (VoD) pair à pair (P2P). L'adaptation de la qualité dans le système de streaming VoD P2P se fait en deux phases.

Tout d'abord en utilisant la stratégie LLI (Layer Level Initialization), une qualité initiale est sélectionnée en fonction des ressources statiques du pair, par exemple la résolution d'écran, la bande passante et la puissance de l'appareil. Après le démarrage du streaming, un autre ensemble d'algorithmes appelé LLA (Layer Level Adjustment) est utilisé pour adapter la qualité vidéo aux différents paramètres dynamiques des pairs (mémoire, consommation d'énergie) et du système (disponibilité des blocs, débit). LLI et LLA sont évalués sur les métriques de qualité de service (QoS) des SVC au niveau de la couche à l'aide des simulateurs Peersim et JSVM. Enfin, nous avons parfaitement assuré la meilleure QoS pour les couches SVC streamables par nos algorithmes proposés qui maximisent la flexibilité SVC tout en prenant en compte l'impact de la consommation de différentes ressources et affecte la résilience des systèmes de streaming P2P.

SVC est plus pratique pour prendre en charge les applications de streaming avec des appareils hétérogènes en termes de bande passante de communication, de résolution d'affichage et de CPU.

LLI est évalué et comparé à IQA sur les métriques QoS du niveau de couche SVC, mais LLA est testé dans l'évaluation préliminaire à l'aide de Peersim et de simulateurs JSVM.

1. Formulation de la solution :

a. Initialisation au niveau de la couche (LLI) :

L'idée principale de LLI est de comparer les exigences de chaque couche du flux vidéo avec les ressources statiques locales d'un pair. La propriété subtile du LLI est qu'il doit prendre une décision sur le niveau de qualité sans avoir d'informations sur le débit et la dynamique du système.

Le module LLI évalue les ressources et les besoins actuels afin de les faire correspondre avec une qualité réalisable. Ce module gère principalement les paramètres statiques, tels que la résolution d'écran, la bande passante, la puissance de l'appareil actuellement disponible (CPU, RAM, autonomie de la batterie) et les préférences de l'utilisateur (résolution d'affichage, fréquence d'images, niveau PSNR).

Une qualité initiale définie avec les paramètres de niveau de qualité de couche de base d_0 , t_0 , et q_0 est peuplée au début. Puis, l'évolutivité spatiale, scalabilité temporelle, débit, scalabilité de la qualité, complexité adaptation et distribution longueur de la vidéo durée de vie relative de la batterie les modules d'adaptation sélectionnent tous les niveaux de qualité compatibles basés sur la résolution de l'écran, la bande passante et l'alimentation du périphérique respectivement en considérant les limitations de préférence utilisateur. Toutes les combinaisons compatibles sont ajoutées en tant que candidats. Le final la décision est prise en sélectionnant l'ensemble de qualité filtré $L(d, t, q)$ qui les valeurs des trois dimensions sont maximales. Le LLI décision définitive est de préférer les articles avec une valeur SNR plus élevée.

L'architecture du LLI est illustrée à la figure 2.2

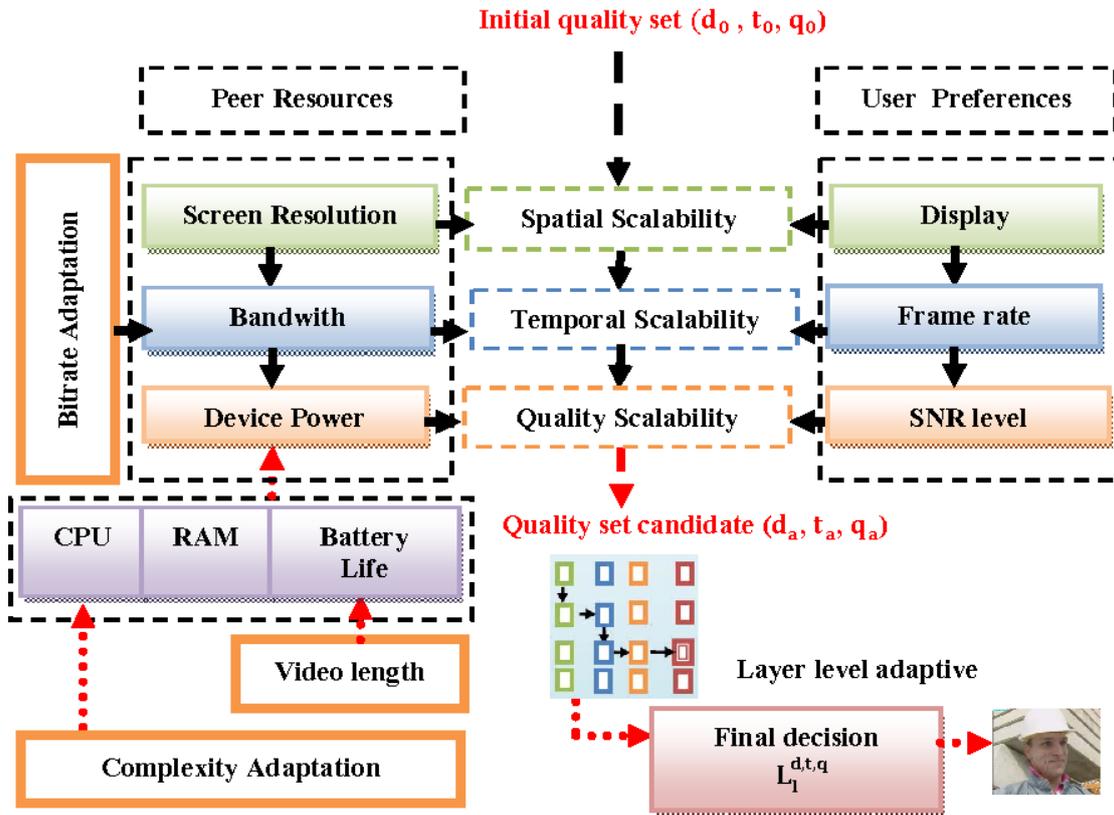


Figure 2.2. Algorithme LayerLevelInitialization (LLI)

b. Réglage du niveau de couche (LLA)

Le LLA est exécuté périodiquement pendant le streaming dans le cadre d'une boucle de contrôle pour garantir une adaptation en douceur. Ce module s'adapte aux changements des

conditions du réseau afin de maximiser la qualité disponible au niveau du récepteur. Étant donné que la taille d'écran de l'utilisateur ne change pas pendant la lecture vidéo, seules les adaptations temporelles et SNR sont gérées par le LLA car généralement la résolution d'affichage des pairs est inchangeable. La couche spatiale ne sera pas modifiée par le LLA.

Le LLA utilise des informations en temps réel sur l'état du réseau mesurées par la disponibilité des blocs dans la taille du partenariat et le débit de téléchargement actif. Il prend la couche SVC actuelle adaptative en entrée, puis l'ajuste en fonction des informations réseau en temps réel.

Cette couche est traitée par les différentes étapes de la LLA pour produire une nouvelle couche qui correspond aux conditions actuelles du réseau. Par conséquent, le LLA part du préfiltré des paramètres de niveau de couche de sortie LLI $\{d, t, q\}$. Les quatre étapes d'adaptation du LLA forment ensemble le processus de prise de décision.

L'état du réseau, le débit binaire, la complexité et les composants d'adaptation du niveau PSNR ajustent tous les niveaux de qualité compatibles en fonction de la disponibilité des blocs, du débit et de la puissance de l'appareil (CPU, RAM, autonomie de la batterie) respectivement. Ici, l'indicateur de disponibilité de bloc fournit des informations sur les couches disponibles dans le système P2P. L'architecture du LLA est illustrée à la figure 2.3

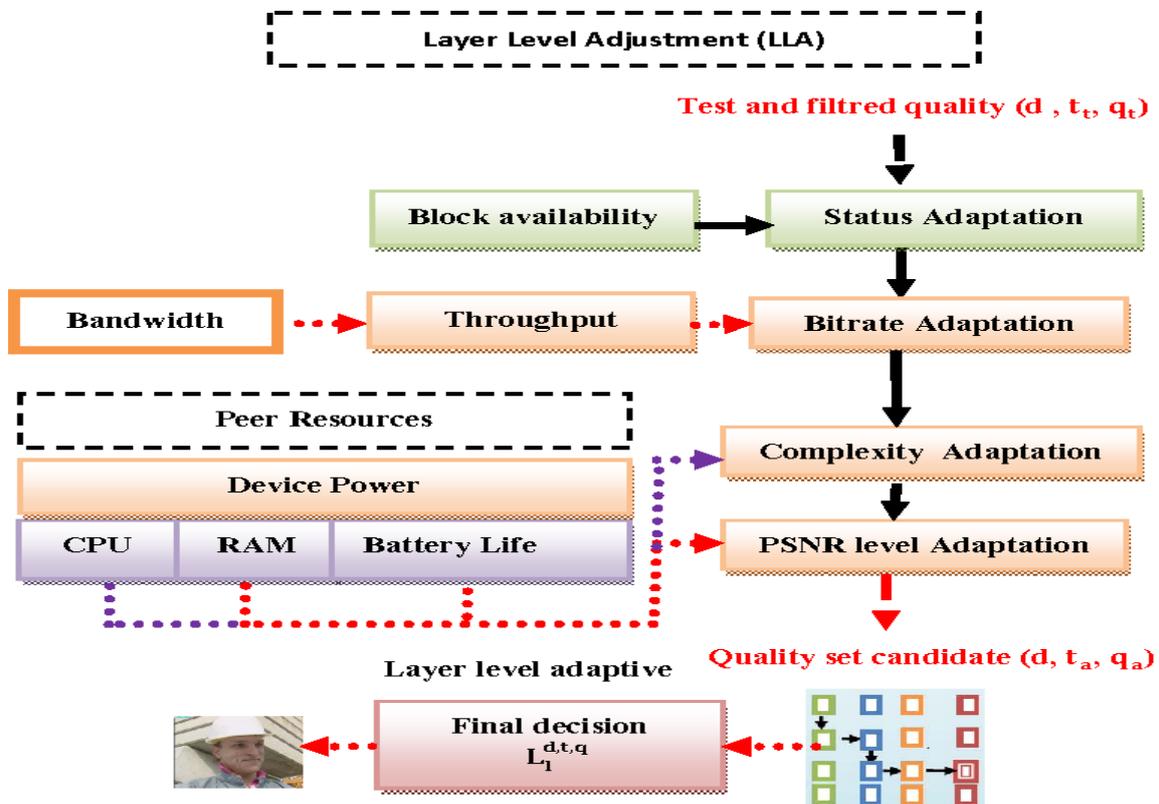


Figure 2. 3.AlgorithmeLayer Level Adjustment (LLA)

2. Les avantages de l'approche

- Appliqué à un service de streaming de bout en bout (p2p).
- Consommation de toutes les ressources de l'utilisateur afin de maximiser la qualité du flux.
- Utilisation de plusieurs métriques.

3. Les inconvénients de l'approche

- L'architecture de l'approche est décentralisée.
- Sa complexité est haute.
- Basé uniquement sur l'amélioration de la qualité du service et non sur la qualité d'expérience en général.

2.2.3 MS-Stream : une solution de streaming adaptatif multi-sources améliorant la qualité perçue des consommateurs [4]

Afin d'offrir la possibilité d'exploiter une plus grande largeur de bande, la diversité des liens et la fiabilité et ça grâce à l'utilisation de plusieurs serveurs, la solution MS-Stream est proposée comme une évolution des protocoles de streaming adaptatifs actuels.

En apportant un support multi-sources aux solutions de streaming adaptatif basées sur HTTP existantes, un nouveau type d'application de streaming multimédia est développé. La figure 2.4 présente un aperçu de haut niveau de la solution MS-Stream, où toute source de contenu peut contribuer aux sessions de streaming, en fonction de son type de connectivité réseau.

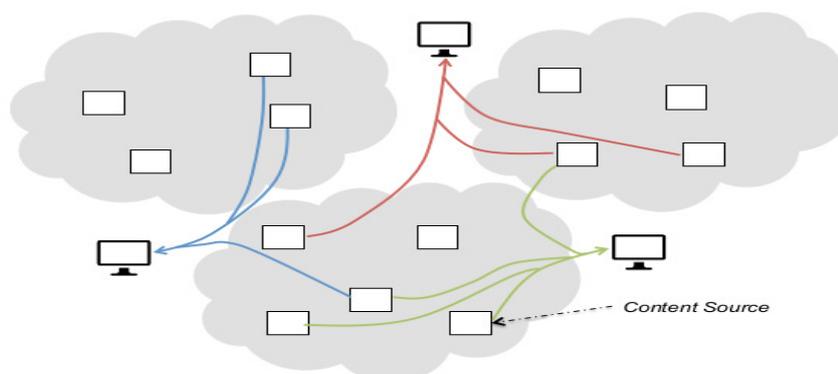


Figure 2. 4. Présentation de la solution MS-Stream

a. formulation de la solution :

Les éléments clés de la proposition résident dans les innovations apportées au schéma de codage/décodage et au protocole de diffusion en continu, ainsi qu'à l'algorithme qui lui est associé. Cette contribution peut donc être décrite par les deux sections suivantes :

- le schéma de génération de sous-flux conforme à la norme qui sera utilisé par le protocole MS-Stream.
- le protocole MS-Stream lui-même avec son adaptation de description, son mécanisme de commutation de serveur et son algorithme de commutation intra-description.

1. Le schéma de génération de sous-flux proposé :

Le schéma de génération de sous-flux utilisé dans cette approche s'appuyant sur les principes mis en évidence par le codage de description multiple (MDC). Chaque vidéo est générée sur plusieurs sous-flux conformes aux normes, appelés descriptions. Le nombre de descriptions, pour un flux donné, est calculé avant la phase de diffusion.

La qualité vidéo augmente avec le nombre de descriptions fusionnées et la qualité originale peut être obtenue lorsque toutes les descriptions sont utilisées. Ces descriptions sont indépendantes les unes des autres, ce qui est rendu possible par la copie de certaines informations communes (c'est-à-dire la redondance) dans toutes les descriptions. La non-dépendance entre les flux assure une grande fiabilité dans les réseaux hétérogènes ou peu fiables, car tout sous-flux peut être perdu sans interrompre la session de diffusion en continu. Plus la redondance est importante, plus la consommation de bande passante du réseau est élevée. Par conséquent, et afin de répondre à tout scénario spécifique (par exemple, de sources fiables à haut débit à des sources volatiles à faible débit), MS-Stream dispose de mécanismes intégrés pour contrôler le degré de redondance en fonction de la disponibilité des serveurs, de l'état du réseau et de la qualité demandée par le client.

Contrairement aux autres approches MDC proposées, cette solution peut générer un nombre illimité de descriptions, avec l'utilisation des deux domaines spatial et temporel de MDC. À partir de leurs travaux antérieurs, ils proposent une stratégie de codage à descriptions multiples à plusieurs débits basée sur le groupe d'images (GOP). La création de description est le résultat d'une étape de post-codage, qui entrelace les GOP à partir de deux représentations de débit binaire du même contenu, comme le montre la Figure 2.5.

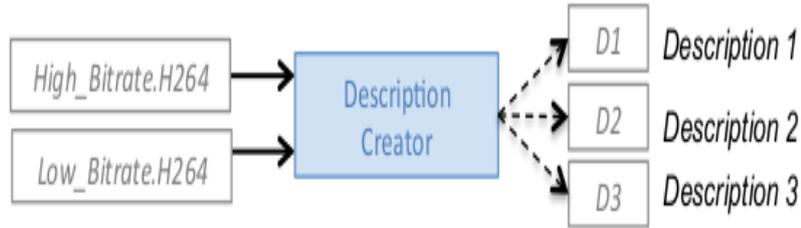


Figure 2. 5. Processus de génération de descriptions multiples à partir du contenu H.264

Et des exemples de descriptions basées sur GOP sont illustrés à la Figure 2.6.

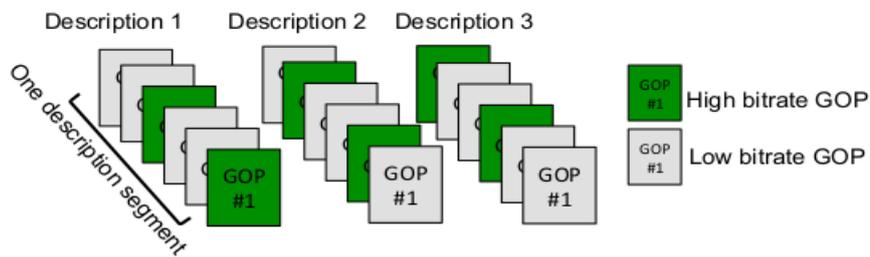


Figure 2. 6. Exemple de répartition GOP pour créer 3 descriptions différentes

Une étape de pré-décodage peu complexe est nécessaire pour fusionner les descriptions. Reconstruire le contenu original à partir des descriptions générées est donc une simple question de ré-ordonnancement des données, en sélectionnant les GOP de taille supérieure dans le pool de descriptions disponibles (Figure 2.7). Si toutes les descriptions ne sont pas disponibles pour la reconstruction du contenu côté client, le contenu reste lisible avec une qualité visuelle non optimale, en raison d'une certaine perte d'informations au niveau des GOP.

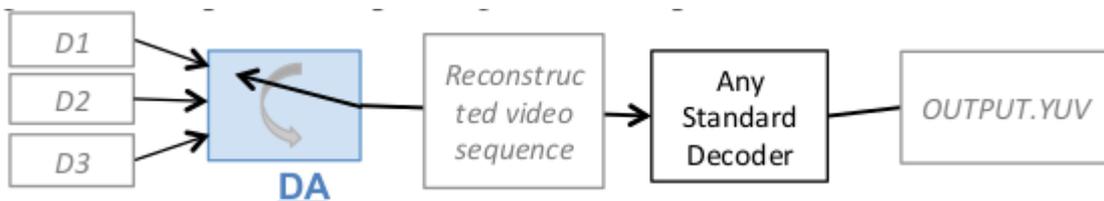


Figure 2. 7 Décodage de descriptions multiples avec l'agrégateur de description (DA)

Donc, finalement, on peut conclure que ce schéma de génération de sous-flux a donné les moyens de créer des descriptions indépendantes à placer dans plusieurs serveurs sources.

2. Le protocole MS-Stream et les algorithmes associés :

Un protocole de streaming évolutif est proposé : Streaming multi-source sur HTTP (MS-Stream), où les aspects côté client sont conservés et une utilisation pragmatique de la bande passante réseau sur plusieurs chemins est suggéré :

2.1 Protocole de streaming multiserveurs :

En récupérant plusieurs descriptions de plusieurs serveurs à la fois, la probabilité de recevoir au moins un flux est augmentée et une lecture fluide du contenu est assurée. Pour atténuer le problème d'épuisement du contenu mis en mémoire tampon disponible lorsque la bande passante du réseau diminue fortement (de 6 Mbps à 1 Mbps), ce qui peut entraîner un réamorçage, une surveillance de la progression du téléchargement et un abandon éventuel du téléchargement d'un segment de description sont utilisés.

Deux mesures sont utilisées pour définir un délai d'expiration défini sur chaque demande : contenu mis en mémoire tampon et durée de lecture de segment par défaut.

- ➔ Si le contenu mis en mémoire tampon peut assurer la diffusion du contenu pendant moins de deux fois la durée de diffusion du segment par défaut, alors le délai d'attente est fixé à 90 % de la durée de diffusion du morceau. Sinon, aucune temporisateur n'est définie sur la demande de segment.
- ➔ Dans le cas où toutes les descriptions en cours d'une demande de segment durent plus longtemps que le time-out défini, les demandes sont maintenues en vie jusqu'à ce qu'une description soit téléchargée.
- ➔ Si la diffusion du contenu mis en mémoire tampon disponible est inférieure à la moitié de la séquence de durée de segment par défaut et qu'une description a déjà été reçue, les autres téléchargements de descriptions complémentaires en cours sont abandonnés afin d'éviter la mise en mémoire tampon.

2.1.1. Adaptation de la description :

Lors de la réception des descriptions des segments, les clients MS-Stream estiment la largeur de bande disponible sur le chemin utilisé pour une description donnée et par conséquent, ils peuvent augmenter ou diminuer la représentation du débit binaire de la prochaine description du segment à télécharger. Cette fonction d'adaptation de la description de MS-Stream répond à trois problèmes différents :

- Si un client télécharge une description de segment sur un chemin de réseau en mauvais état, la probabilité de recevoir ce dernier segment périmé est élevée. MS-Stream atténue ce problème en laissant deux choix aux clients : Récupérer le segment de description suivant avec un degré de redondance plus faible, ou ils peuvent décider de récupérer les segments de description complémentaires suivants à partir d'autres serveurs) avec un degré de redondance plus élevé, afin de compenser la perte de la description identifiée.
- Le deuxième problème traité par l'adaptation de description MS-Stream consiste à minimiser la surconsommation de bande passante réseau provenant du codage de descriptions multiples. Ce problème n'est pas directement lié à la QoE du consommateur mais plutôt au côté serveur des contraintes qui, à son tour, impactent la QoE du consommateur. Par conséquent, lorsqu'un état stable est atteint dans une session de streaming où les descriptions sont toujours reçues à temps, les clients MS-Stream ont la possibilité de modifier les demandes de segment afin de réduire la quantité de redondance par description (c'est-à-dire jusqu'à une surcharge presque nulle).
- Le dernier problème est lié à la fourniture d'une qualité de contenu optimale tout en assurant une lecture vidéo fluide en augmentant et en diminuant la qualité en fonction du débit observé.

2.1.2. Mécanisme de commutation du serveur

Le protocole MS-Stream offre également aux clients la possibilité d'effectuer simplement des opérations de commutation de serveur pendant la session de streaming. Deux cas différents sont traités par ce mécanisme d'adaptation de serveur :

- Si la capacité de téléchargement du serveur utilisé par le client devient insuffisante pour livrer les descriptions demandées dans la qualité souhaitée. Le passage à un autre serveur permettrait à ces clients de recevoir éventuellement les descriptions des segments souhaités.
- Si les clients ne pouvaient trouver que des serveurs appropriés pour récupérer des segments de description dans une qualité sous-optimale, les clients MS-Stream ont la possibilité d'effectuer 'server handover' qui est un atout important pour éviter les situations de stagnation et pour donner aux clients la possibilité de trouver des serveurs et des chemins réalisant débit plus élevé.

```

Input : S, s
1.  Tant que la session de streaming n'est pas terminée
2.    téléchargement De Segment (s, T, Q)
3.    Mettre à jour l'historique du serveur ()
4.    Si s servi avec succès les N derniers segments
    Qualité Q Alors
5.      Si Augmenter Décision qualité () est vrai Alors
6.        Q = Q+1
7.      Sinon
8.        Si (Q < Qmax for the last M segments) Alors
9.          s ← effectuer ServerHandover ()
10.       Sinon
11.        Si qualité Q a été testé sur tous les serveurs de S Alors
12.          Si Q != 0 Alors
13.            Q ← Q-1
14.          Sinon
15.            s ← effectuerServerHandover ()
16.        (T, N, M) ← RecalculerTNM (Q, s)
    
```

2.1.3. Algorithme de commutation intra-description : Afin de fusionner les décisions d'adaptation de la description et de commutation de serveur, nous proposons l'algorithme de décision suivant 1. L'objectif de l'algorithme est de décider s'il faut procéder à l'adaptation de

la description ou à l'adaptation du serveur pour une description à la fois et comment le faire (d'où le nom "intra-description switching"). L'algorithme démarre avec les informations disponibles dans le fichier MPD qui contient une liste de serveurs associés aux descriptions hébergées.

L'algorithme vise, par ordre de priorité :

- i. en conservant un serveur qui fournit la qualité de description demandée (ligne 4),
- ii. en augmentant la qualité de la description demandée, lorsque cela est possible (ligne 5-6),
- iii. le passage à un autre serveur, si le client ne peut pas obtenir la qualité optimale pour une quantité spécifique de demandes (ligne 8-9), la directive 'PerformServerHandover' n'est appliquée que si aucun autre server handover n'a lieu en même temps pour une autre description ;
- iv. en diminuant éventuellement la qualité de la description demandée si et seulement si tous les serveurs disponibles énumérés dans le fichier MPD ont été testés pour la qualité actuelle (ligne 11-12-13).

b. Les avantages de l'approche

- La possibilité pour la plupart des appareils connectés au réseau de faire partie d'une session de diffusion, ce qui donne aux fournisseurs de flux l'avantage de ne plus dépendre entièrement sur des infrastructures de serveurs surdimensionnées et coûteuses (en termes de CPU, de RAM et de bande passante).
- agnosticisme codec et rétro conformité DASH.
- Faible complexité supplémentaire de codage/décodage.
- La dispersion du trafic et donc réduit la congestion du réseau.
- Le codage de description multiple utilisé (MDC) est apparu comme une approche appropriée pour améliorer la résilience à l'erreur d'un système de diffusion vidéo.
- Applicable à tous les modèles d'architecture de streaming (P2P, CDN, Clouds), ce qui permet d'obtenir une diffusion de contenu multiserveurs légère et facile à concevoir.
- Solution décentralisé donc il n'y a pas de véritable point de défaillance unique.

c. Les inconvénients de l'approche :

- Plus la redondance est importante, plus la consommation de bande passante du réseau est élevée.
- Cet overhead a un impact direct sur la capacité de mise à l'échelle du système de streaming pour accommoder le plus grand nombre possible de clients.

2.2.4 Étude comparative

Pour analyser et à synthétiser les points communs, les différences et les tendances de ces 3 solutions envisageables pour une adaptation de la vidéo optimisée ainsi une amélioration de QoE des clients, on a établi l'étude comparative suivante :

Notre étude est basée sur les trois approches dans [25], [26] et [4], dont l'une est plus complexe que l'autre. Le schéma proposé dans [25] adapte la qualité vidéo en tenant compte à la fois de la variation de la largeur de bande du réseau et de l'occupation de la mémoire tampon observée du client. Cependant, l'autre solution [26] plus compliquée lorsque elle s'appuie sur l'utilisation efficace des ressources diverses disponibles de chaque homologue afin de maximiser la qualité du flux délivré, et finalement on a la proposition MS-Stream [4] qui utilise un adaptatif protocole de streaming qui dépend sur deux mécanismes d'adaptation : un sous-flux d'adaptation et un serveur d'adaptation.

Ainsi, chaque approche possède son but pour offrir le client avec une qualité vidéo optimale. Pour réaliser les résultats souhaités les solutions proposées dépendent sur les métriques montré dans le tableau suivant :

Approches	Buts	Améliorer QoE/QoS	Métriques	Evaluation
[25]	Diminuer le nombre de changements de qualité vidéo et pour atténuer le dépassement de la mémoire tampon tout en maintenant le débit moyen comparable à la vidéo conventionnelle programme d'adaptation de la qualité.	QoE	-Evénements de changement de qualité vidéo. - Moyenne débit. - Occupation du tampon.	Simulation
[26]	Ajustement de la qualité vidéo à l'aide de SVC tenant compte de la consommation des différentes ressources et la résilience des systèmes de streaming P2P.	QoS	-La résolution d'écran, la bande passante, CPU, RAM. - Consommation d'énergie, disponibilité des blocs, débit.	Simulation
[4]	Amélioration de la qualité par l'utilisation de plusieurs serveurs la diversité des chemins d'accès.	QoE	-Nombre d'événements de remis en mémoire tampon. - Distribution de qualité tout au long de la session de streaming. - Evénements de changement de qualité vidéo. -Amplitude moyenne de changement de qualité. -Débit binaire moyen du contenu affiché.	Prototype en ligne

Tableau 2. 3 Comparaison générale entre les approches

Chaque approche diffère de l'autre, chacune ayant ses avantages et ses inconvénients et ça rend la comparaison entre eux plus compliquée, mais finalement nous pouvons conclure les points de similitude et les points de différence suivant :

➤ **les similitudes :**

- [25], [26], [4] considèrent la bande passante comme un facteur pour sélectionner le bon débit binaire de la vidéo.
- la sélection de segment/layer/description dans [25], [26] et [4] est effectuée par client au lieu des serveurs de streaming.
- [26] et [4] utilisent des sous-flux pour la diffusion qui ont appelé 'layers' dans [26] et 'descriptions' dans [4].
- [25] et [4] exploite tous les deux une architectures client / serveur.

➤ **Les différences :**

- La dépendance entre les couches dans [26] rend la programmation des segments complexe et si l'on ne parvient pas à récupérer la "couche de base", le consommateur ne pourra pas regarder la vidéo et ça contrairement à MS-Stream dans [4] qui n'a pas cette limitation car chaque description est indépendante de l'autre donc en cas de perte des descriptions, il reste la possibilité de regarder la vidéo demandée.
- L'approche dans [26] utilise plusieurs paramètres par rapport aux autres, ce qui est un avantage pour sa faveur.
- L'approche [26] prend en compte l'évaluation en utilisant les métriques de la QoS au contraire des approches dans [25] et [4].
- Les approches dans [25] et [26] utilisent la simulation tandis que l'approche dans [4] utilise Un prototype en ligne.
- [26] spécifie la vidéo à la demande comme un type de streaming autre que [25] [4] ne le spécifie pas.

2.3 Conclusion

Dans ce chapitre, trois approches différentes ont été discutées pour rassembler des données afin d'évaluer et améliorer la QoE dans les services de vidéo streaming et analyser l'impact de différents paramètres .Puis, on a décrit les avantages et les inconvénients de chacune. On a terminé ce chapitre par une étude comparative.

Le chapitre suivant est réservé à la conception d'une approche qui vise à résoudre un problème prédominant et adressé par les approches d'amélioration de la qualité d'expérience des utilisateurs discutées dans ce chapitre. Notre approche sera basée sur un processus de révision de ce qui a été téléchargé, et cette idée n'a pas été adoptée par d'autres approches.

**CHAPITRE 3 : Conception d'un mécanisme
d'amélioration de la QoE dans les services de vidéo
streaming adaptatif HTTP**

3.1 Introduction

Dans ce chapitre, nous décrivons notre solution qui vise à améliorer la qualité d'expérience dans les services de vidéo streaming adaptatif.

Au cours de ce chapitre, nous aborderons les points suivants : le principe de solution proposée, la description, l'algorithme utilisé et enfin un organigramme illustre brièvement le mécanisme utilisé.

3.2 La solution

3.2.1 Principe de la solution

Afin d'améliorer la qualité d'expérience des services de vidéo streaming adaptatif, plusieurs méthodes conventionnelles et méthodes récentes essaient d'apporter des changements sur la politique d'adaptation. De bon résultat dans l'amélioration de la qualité d'expérience de l'utilisateur ont été obtenu.

La solution proposée améliore la qualité vidéo quel que soit la stratégie d'adaptation. Elle modifie le contenu de la mémoire tampon qui contient les segments vidéo téléchargés.

Afin d'améliorer la qualité d'expérience des utilisateurs (améliorer leur satisfaction vis à vis du service de streaming), notre solution va agir sur le contenu du buffer de lecteur pour maximiser la qualité vidéo reçue et de minimiser le changement de qualité au cours de la diffusion du média.

3.2.2 Description

Avant de décrire la solution et son fonctionnement, nous présenterons la politique d'adaptation utilisée pour remplir notre buffer avec des segments vidéo de qualité appropriée avec la bande passante disponible .Et puis nous détaillerons notre solution « révision du flux ».

3.2.2.1 Adaptation

Traditionnellement, la plupart des mécanismes d'adaptation de la qualité vidéo de la diffusion en continu adaptative HTTP mesurent le débit des segments pour une diffusion vidéo continue chaque fois que chaque segment est arrivé au client.

Nous utilisons au cours de ce travail une méthode d'adaptation pour sélectionner la meilleure qualité vidéo conforme à la bande passante disponible. La méthode adoptée peut être illustrées comme suit :

1. Estimation de la bande passante :

Après le téléchargement de chaque segment, nous mesurons le débit (DBS[i]) de ce dernier en calculant la quantité des octets téléchargés (BitD) et en divisant cette quantité par le temps nécessaire pour télécharger du segment (T) comme on peut le voir dans l'équation (1). Au final nous apprécions la bande passante disponible en calculant la moyenne (**MDébits**) de tous les débits des segments passés comme le montre l'équation (2).

$$DBS[i] = \text{BitD} / T. \quad (1)$$

$$\text{MDébits} = \sum DBS[i] / n. \quad (2)$$

Avec n : nombre des segments téléchargés.

2. Adaptation de la qualité vidéo :

En utilisant la valeur de la bande passante estimée précédemment, nous la comparons à tous les débits de qualité vidéo disponibles et sélectionner par la suite la qualité vidéo la plus élevée sous cette valeur estimée. De cette façon, nous obtenons la meilleure qualité vidéo disponible.

3.2.2.2 Révision du flux

En raison de la perturbation de la bande passante, il est possible à un moment donné de télécharger un segment vidéo de mauvaise ou de haute qualité. Ainsi, plusieurs changements de qualité peuvent être produits et cela dérangera l'utilisateur. Donc pour éviter ce problème, nous essayons de remplacer ces segments faibles ou forts par des segments alternatifs de qualité fréquente dans notre buffer

Alors, la solution est composée de plusieurs étapes :

a. Trouver la qualité (bitrate/résolution) d'occurrence maximale :

Après avoir téléchargé un certain nombre de segments vidéo encodés dans la mémoire-tampon et atteint la taille de la mémoire-tampon à un certain seuil, nous calculons

l'occurrence de chaque qualité disponible de la vidéo donnée et retournons la qualité d'occurrence maximale.

b. Extraire les segments vidéo de mauvaise ou bonne qualité :

Afin de trouver les segments vidéo indésirables qui peuvent provoquer un changement de qualité gênants, nous parcourons le tampon pour extraire les numéros des segments vidéo dont ils ont une qualité inférieure ou supérieure à la qualité maximale.

c. Télécharger les segments alternatifs :

À ce niveau, les segments marqués seront téléchargés dans un tampon séparé, et cela pour éviter de perdre le segment vidéo (original) après l'avoir supprimé pour placer l'alternatifs et l'échec de télécharger ce alternatif, donc l'impossibilité de continuer la lecture de la vidéo.

d. Remplacer les segments indésirables :

Une fois le téléchargement des segments avec la qualité approprié est terminé nous parcourons le tampon de téléchargement pour remplacer faible/forts segments. Et en cas d'un segment à remplacer n'existe plus, le segment remplacé sera automatiquement supprimé. Et la figure (3.1) illustre l'opération de remplacement du segment indésirable avec un autre similaire avec la bonne qualité.

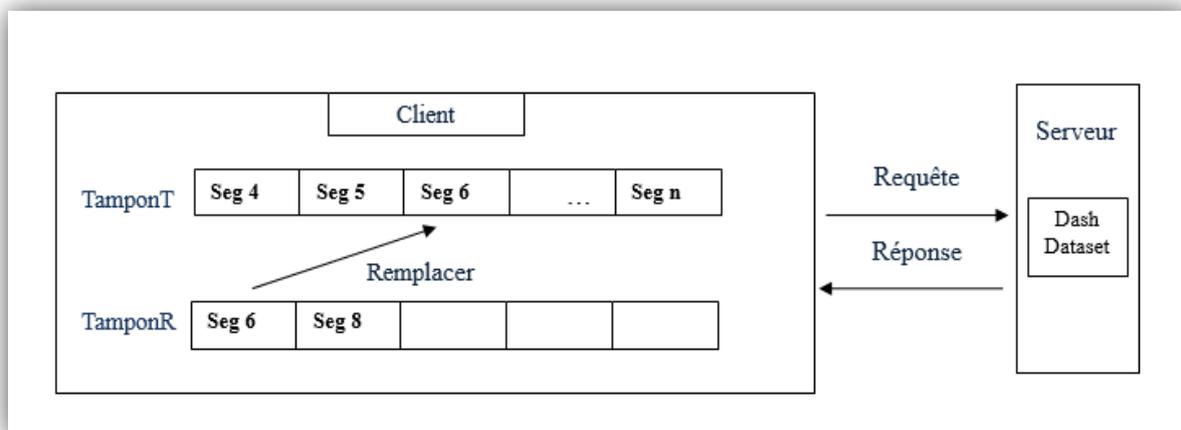


Figure 3. 1. Remplacement des segments

3.2.3 Les paramètres initiaux

Pour démarrer l'opération de révision du flux vidéo, nous devons prendre certains points en considération :

- en raison de la logique d'adaptation utilisée qui calcule le débit binaire moyen dès le début de la lecture, cela signifie qu'il faudrait beaucoup de temps pour répondre aux fluctuations de la bande passante pendant la lecture [14] (lorsque la bande passante du réseau diminue considérablement). cela peut provoquer sous-dépassement de buffer et des événements de rebuffering si le buffer n'est pas assez plein.
- La révision doit se faire après le téléchargement d'un certain nombre de segments pour remplir le tampon avec des segments représentant la bande passante disponible.
- La révision ne doit pas être sur les premiers segments car ils sont proches de la tête de lecture.

La méthode débute avec ces paramètres suivants :

- **SR** : seuil pour commencer la révision.
- **NSR** : nombre segments téléchargés avant la révision.
- **SP** : nombre des premiers segments invérifiable.

3.2.4 Algorithme de la solution

Dans cette section nous allons présenter l'algorithme de notre solution. Le tableau des variables utilisées est le suivant:

Symbole	Description
tamponT []	Tampon du téléchargement de taille m.
tamponR []	Tampon r des segments alternatives de taille m.
TableQ []	Tables des bitrates et les résolutions correspondent disponibles de taille f.
Avg	Moyenne des débits des segments passés.
NST	Nombre de segments à télécharger avant de la révision.
BitD	Les bits téléchargés.
VNS []	Tableau contient les numéros des segments pour les remplacer.
BDA	Bande passante actuelle.
N	Nombre de segments téléchargés.
AQ	Variable pour activer ou désactiver l'opération de vérification.
Q	Qualité actuelle.
Qmax	Qualité d'une grande occurrence dans buffer1.
Url	Url de fichier mpdd'une vidéo.

Tableau 3. 1. Tableau des variables

Dans ce qui suit, nous exposerons l'algorithme utilisé pour la révision et l'algorithme utilisé pour l'adaptation.

Algorithme de la révision du flux vidéo :

Var :

TamponT [], tamponR [] : segment vidéo ;

URL : chaine de caractères ;

*TableQ [], Q, Qmax: **Qualité** ;*

*BitD, nbrS, VNS, N: **entier** ;*

*BDA, AVG : **réel** ;*

*AQ : **booléen** ;*

1. Début

2. *Télécharger mpd (URL) ; // télécharger fichier*

3. *Session ← vrai ;*

4. *Q ← TableQ [0] ;*

5. *Tant que (session) faire*

6. *Téléchargement_segment (Q) ; // segment avec qualité Q*

7. *nbrS ← nbrS + 1 ;*

8. *N ← N + 1 ;*

9. *T ← temp_du_téléchargement () ;*

10. *BitD ← quantité_bit_téléchargés () ;*

11. *BDA ← BitD / T ;*

12. *Avg ← (Avg * N + BDA) / N + 1 ;*

13. *Q ← adaptation (Avg, TableQ) ; // sélection qualité suivante selon moyenne du débit desseg*

14. *Si (nbrS > NSRet AQ = vrai et plus_de_SR (tamponT)) alors*

15. *nbrS ← 0 ;*

16. $Q_{max} \leftarrow \text{qualité_avec_max_occurrence}(\text{tamponT});$
 17. **Pour** $i \leftarrow SP$ à **Taille_tamponT** faire // Laisser les SR segments premiers proches de la lecture
 18. $VNS \leftarrow \text{est_un_seg_faible_ou_fort}(\text{tamponT}[i], Q_{max});$ // retourner numéro du seg
 19. **Fin Pour**
 20. $\text{Télécharger_seg_alternatives}(\text{tamponR}[], VNS[]);$
 21. **FinSi**
 22. **Si** (! est_videramponR[]) alors
 23. $\text{remplacer_seg}(\text{tamponT}[], \text{tamponR}[], VNS);$ // remplacer selon numéros des seg
 24. **FinSi**;
 25. **FinTantque**;
 26. **Fin**
-
-

Algorithme de la fonction d'adaptation de la qualité vidéo utilisée :

//La qualité vidéo la plus élevée sous la bande passante estimée est sélectionnée

Entrées :Avg, Q_{max}

1. **Pour** $i \leftarrow 0$ à **taille_TableQ** faire
 2. **Si** ($\text{tableQ}[i] > \text{Avg}$) alors
 3. **Si** ($i = 0$) alors
 4. $\text{Retourner}(\text{tableQ}[i]);$
 5. **Sinon** $\text{Retourner}(\text{tableQ}[i-1]);$
 6. **FinSi**
 7. **Fin pour**
-

- **La complexité de l'algorithme :**
n : Nombre de segments téléchargés
m : Taille de buffer

Donc dans les pires des cas nous avons ce qui suit :

- Pour télécharger tous les segments de la vidéo on a la boucle tant que d'un nombre d'itération **n**
- pour calculer la moyenne des débits des segments passés on a la fonction « somme_élément » qu'elle va additionner tous les éléments (**n**) de la vecteur VB.
- pour trouver la bonne qualité suivante la fonction d'adaptation parcourir la tableQ **f** fois pour trouver le meilleur bitrate sous la bande passante estimé.
- la fonction « qualité_avec_max occurrence » utiliser 2 boucle, une de **m** itérations et l'autre de **f** itération pour calculer l'occurrence de **f** qualité possible pour un segment.
- une boucle pour de **m** itérations pour extraire les segments de fort et de faible qualité (AvecSP =0).
- finalement on a la fonction «remplacer_seg » qui va remplacer **m** segments.

➤ **La complexité de l'algorithme avec la partie de l'adaptation :**

Nous avons : $o(n * (n + f + m * f + m + m))$

tant que : $\max(n * n, n * f, n * m, n * m, n * m * f) = \max(n * n, n * m * f, n * m)$ et le tamponT ne peut pas contenir tous les segments de vidéo donc $n > m$ et f est toujours connu tel début alors la complexité de la l'algorithme est $o(n^2)$

➤ **la complexité de l'algorithme sans la partie de l'adaptation :**

Nous avons : $o(n * (m * f + m + m))$ et ainsi que $\max(n * m, n * m * f, n * m) = n * m * f$ donc la complexité de l'algorithme est égale a $o(n * m)$. « **Quadratique** »

3.2.5 L'Organigramme de la solution

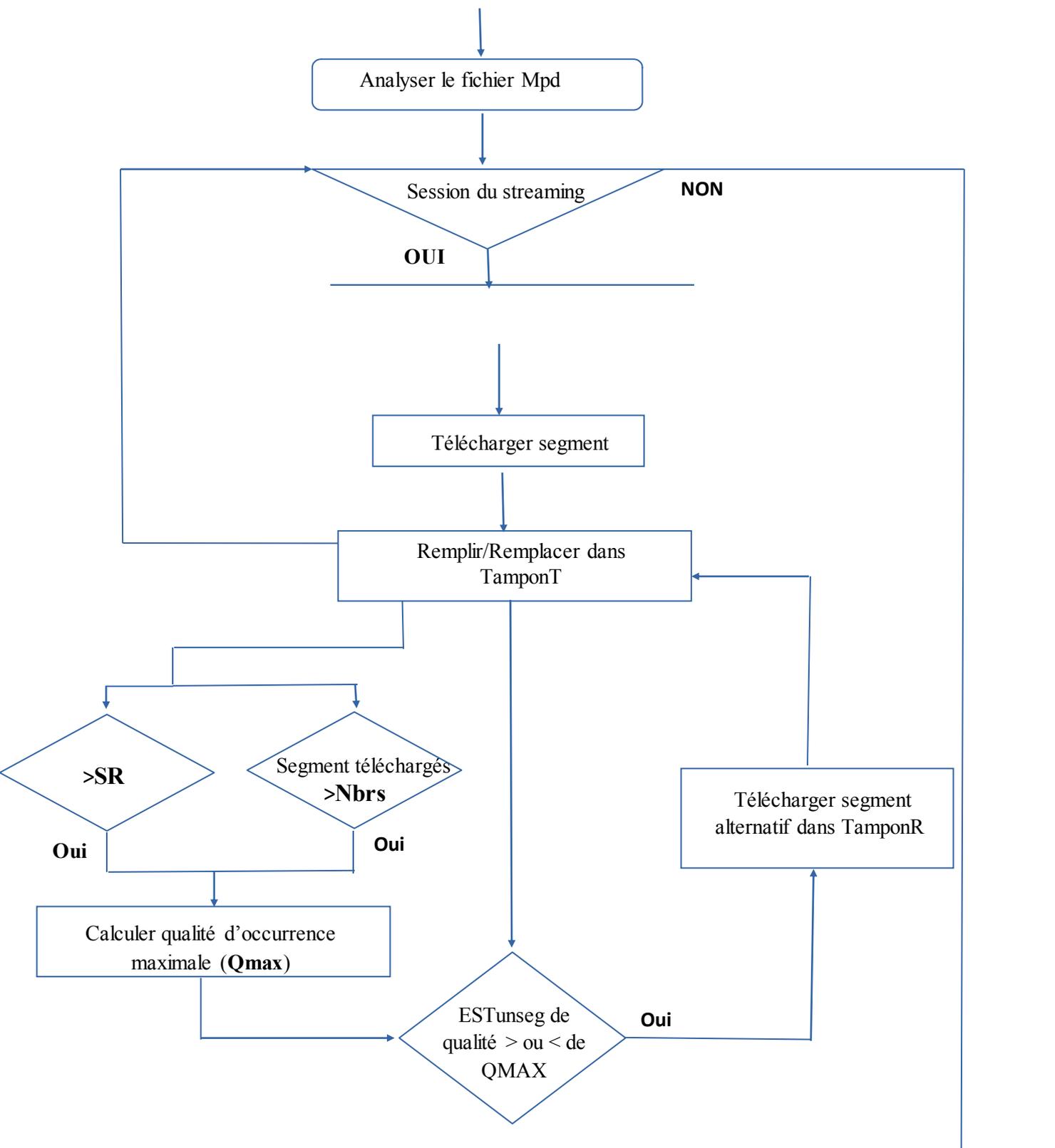


Figure 3. 2. Organigramme de la solution

3.2.6 Conclusion

Nous avons détaillé dans ce chapitre notre solution qui vise à améliorer la qualité d'expérience des services de streaming vidéo adaptatif en agissant sur la mémoire tampon des segments vidéo téléchargés. Le prochain chapitre sera consacré à l'implémentation et à l'évaluation de notre solution, et à la tester par des scénarios pour montrer les points ciblés de la solution.

CHAPITRE 4. Implémentation et Evaluation

4.1 Introduction

Ce chapitre a pour objectif d'implémenter et d'évaluer notre solution en s'appuyant sur une bande passante illimitée. On commence par décrire les outils d'implémentation en montrant la bibliothèque Libdash, et le lecteur Qtsampleplayer et nous terminons par une discussion sur les différents scénarios et résultats d'évaluation de notre solution.

4.2 Outils d'implémentation

4.2.1 La bibliothèque Libdash

L'architecture générale de MPEG-DASH est représentée sur la figure 1 où les parties orange sont normalisées, c'est-à-dire les formats MPD et segment. La livraison du MPD, du contrôle de diffusion DASH, du lecteur multimédia et de l'analyseur de segments est représentée en bleu. Ces pièces ne sont pas standardisées et permettent la différenciation des solutions industrielles, en raison des performances ou des différentes fonctionnalités qui peuvent être intégrées à ce niveau. Notre bibliothèque open source DASH libdash est représentée au niveau du client et encapsule le module d'analyse MPD et HTTP qui est chargé de gérer le téléchargement HTTP par requête via la logique de streaming. Par conséquent, la bibliothèque fournit des interfaces permettant à ces modules d'accéder au MPD et aux segments multimédias téléchargeables. [37]

L'ordre de téléchargement de ces segments multimédias n'est pas géré par libdash et laissé ouvert au contrôle de streaming DASH, qui est un composant indépendant de cette architecture, mais il peut également être inclus dans le lecteur multimédia. Dans un déploiement typique, un serveur DASH fournit des segments dans plusieurs débits et résolutions. Le client reçoit initialement le MPD via libdash, qui fournit alors une interface orientée objet pratique à ce MPD. [37]

Le MPD contient les relations temporelles pour les différentes qualités et segments. Sur la base de ces informations, le client peut télécharger des segments multimédias individuels via libdash à tout moment. Par conséquent, des conditions de bande passante variables peuvent être gérées en passant au niveau de qualité correspondant aux limites des segments afin de fournir une expérience de streaming fluide. Cette adaptation ne fait pas partie de libdash et du standard MPEG-DASH et sera laissée à l'application qui utilise libdash. [37]

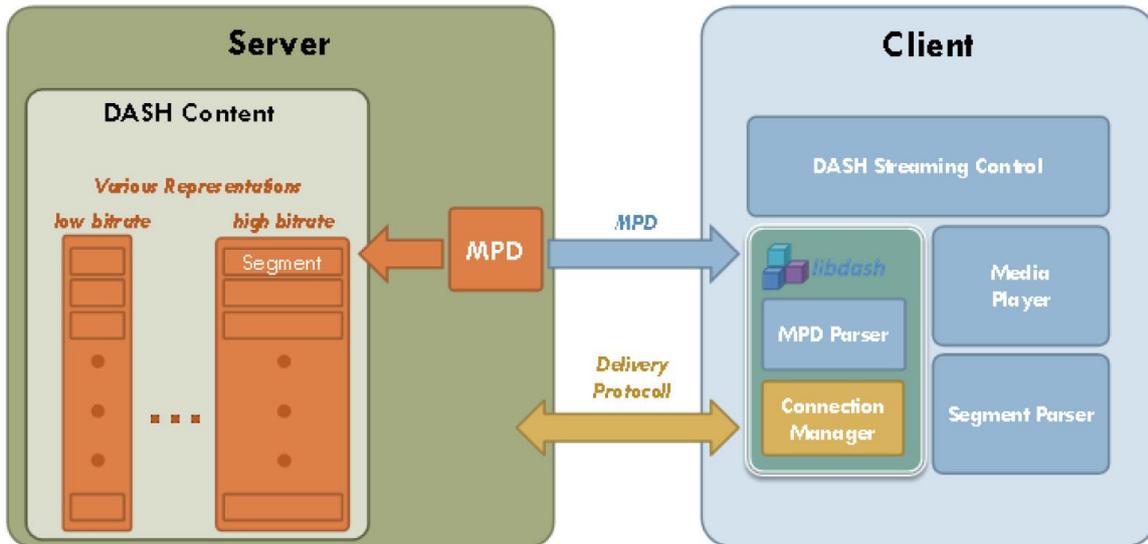


Figure 4. 1. Architecture MPEG-DASH de haut niveau

4.2.2 Le lecteur Qtsampleplayer

Qtsampleplayer est un lecteur DASH complet basé sur libdash et un frontal basé sur QT, il est disponible en open source et fait partie de la suite libdash, qui sont à la fois indépendantes de la plate-forme et fonctionnant sous Windows, Linux, Mac ainsi que les plates-formes embarquées. Avec ce lecteur, différents mécanismes de DASH sont démontrés, par exemple, le contenu sur demande et en direct, les listes de segments, les pistes vidéo et audio séparées ainsi que le contenu multiplexé. [37]

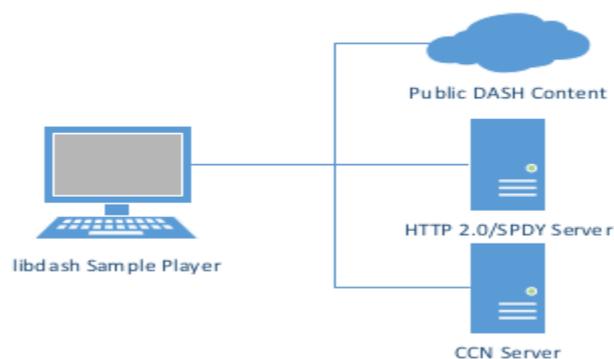


Figure 4. 2. Configuration de la connexion du client

Après avoir téléchargé le code source le libdash, on doit faire une étude sur les différents modules et savoir quels sont les modules qu'on doit changer ou où est ce qu'on doit ajouter des modules.

➤ Interface graphique de notre solution :

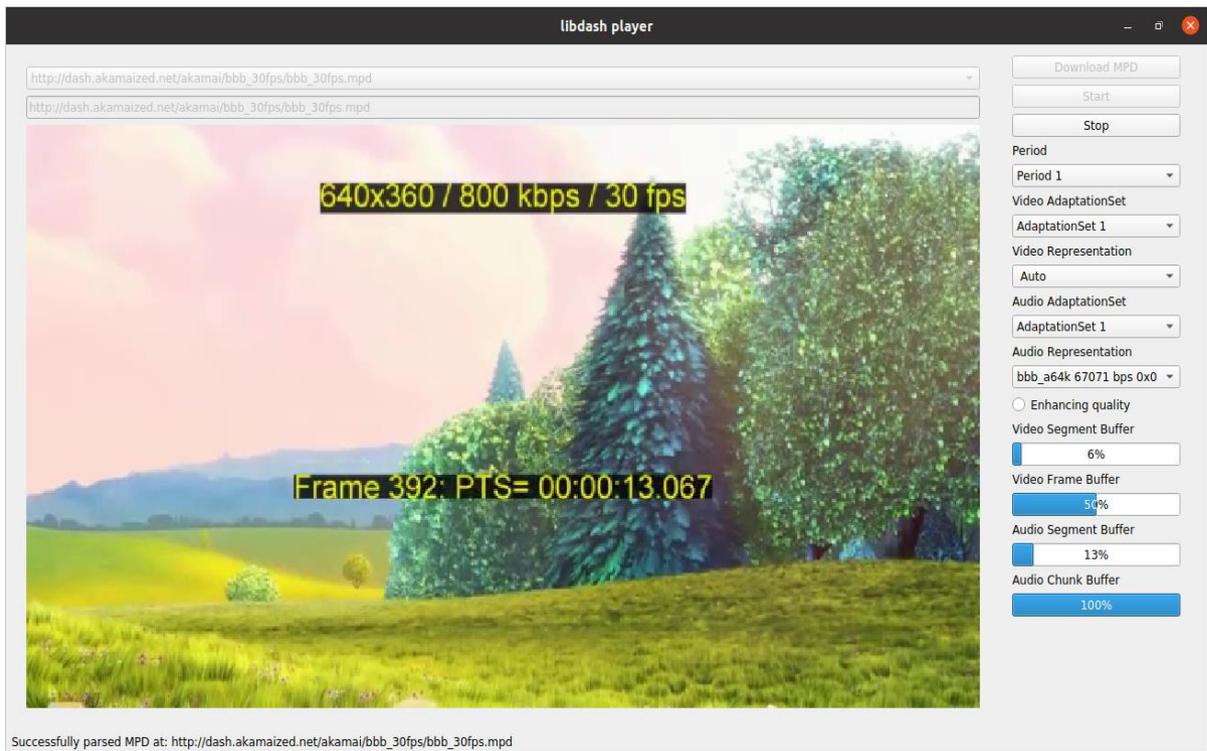


Figure 4. 3. Interface graphique modifiée du lecteur

4.2.3 Matériel

<i>Terminal</i>	<i>Taille d'écran</i>	<i>puissance de Cpu</i>	<i>Capacité de disque dur</i>	<i>Type de Système</i>	<i>Résolution maximale</i>
PC 1	15,6 pouces	2,67 GHz (i5)	320.1 GB	64 bits	1366 x 768
PC 2	15,6 pouces	2,50 GHz (i5)	750 GB	64 bits	1366 x 768

Tableau 4. 1 Paramètres des terminaux utilisés

4.3 Scénario et échantillons du test

Scenarios	But
1	- Observer les changements de qualité qui se produisent pendant la lecture vidéo pour les 2 cas sans et avec révision.
2	- Démontrer que notre solution réussira à remplacer presque tous les segments indésirables.
3	- Voyons le taux de réussite de la révision si nous exécutons une autre application de streaming qui peut rivaliser avec notre lecteur.
4	- Essayer de découvrir l'effet de nos paramètres initiaux choisis.
5	- Proposer de faire une mesure subjective MOS pour tester la perception de la qualité vidéo par les utilisateurs.

Tableau 4. 2. Description des scenarios.

4.3.1 Echantillons utilisés

Dans les expériences suivantes, nous utiliserons des vidéos d'un ensemble de données en ligne "dash.akamaized.net"[23]. Les informations nécessaires sur ces vidéos peuvent être résumées dans le tableau suivant. Ce tableau montre donc tous les niveaux de débit disponibles et les résolutions correspondantes de chaque vidéo ainsi que sa durée et la longueur du segment qui la constitue.

Nom	<i>Big bunny</i>	<i>Envivio - redbull car</i>	<i>Envivio – monsters show</i>
Durée de la vidéo	10:34	4:20	3:14

Longueur du segment	4s	4s	2s
Qualité du segment (bit rate / résolution)	0.25Mb / 320 x 180	0.35Mb / 320 x180	0.2Mb / 320 x 180
	0.5Mb / 320 x 180	0.6Mb / 480 x 270	0.3Mb/ 320 x 180
	0.75Mb / 480 x 270	1.0Mb / 704 x 396	0.48Mb / 512 x 288
	0.8Mb / 640 x 360	2.0Mb / 1024 x576	0.75Mb / 640 x 360
	1.0Mb / 640x 360	3.0Mb / 1280 x 720	1.2Mb / 768 x 432
	1.88Mb / 768 x 432		1.85Mb/ 1024 x 576
	3,13Mb / 1024 x 576		4.3Mb / 1280 x 720
	4,95M / 1280 x 720		5.3M / 1920 x 1080
	9.19M / 1920x1080		
	14.9M / 3840 x 2160		
Nombre de segments	155	65	97

Tableau 4. 3. Les vidéos de test.

4.3.2 Les Scénarios du test

Afin d'évaluer notre solution et de prouver l'amélioration affectée par elle-même, nous avons choisi d'appliquer les scénarios suivants après avoir ajouté les deux nouvelles fonctionnalités, d'adaptation et de révision du flux :

4.3.2.1 Scenario 1:

Dans ce premier scénario, nous simplement apercevons les changements de qualité qui se produisent pendant la lecture vidéo pour les 2 cas sans et avec la révision et l'histogramme suivant représente les résultats obtenus :

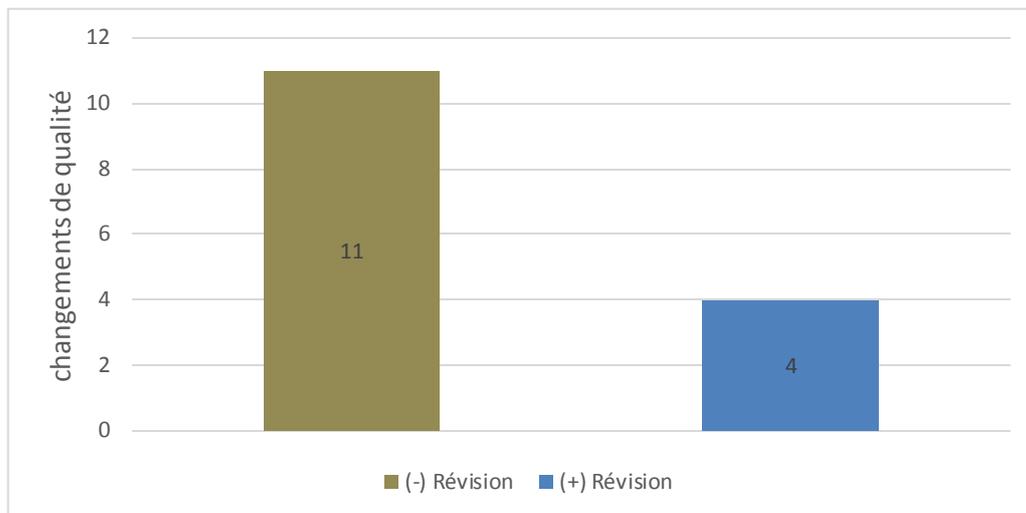


Figure 4.5. Résultats du 1^{er} scénario.

Les deux figures suivantes représentent un exemple du remplacement des segments qu'ils provoquent un changement de qualité :

```

buffer content .....
segment number      Width x height      bitrate(kbps)      add
1                   320 x 180           507                0x7fc90800a670
2                   640 x 360          1254               0x7fc908009aa0
3                   768 x 432          1883               0x7fc90800b960
4                   768 x 432          1883               0x7fc90800b0c0
5                   1024 x 576         3134               0x7fc908001ca0
6                   1024 x 576         3134               0x7fc908002660
7                   1280 x 720         4952               0x7fc908004b10
    
```

Figure 4. 6. Exemple de contenu du tampon avant la révision.

```

buffer content .....
segment number      Width x height      bitrate(kbps)      add
2                   640 x 360          1254               0x7fc908009aa0
3                   768 x 432          1883               0x7fc90800b960
4                   768 x 432          1883               0x7fc90800b0c0
5                   768 x 432          1883               0x7fc908003250
6                   768 x 432          1883               0x7fc908003690
7                   768 x 432          1883               0x7fc908003b80
    
```

Figure 4. 7. Exemple de contenu du tampon après la révision.

Discussion :

- Malgré nous utilisons un algorithme d'adaptation qui réduit à son tour les changements de qualité qui se produisent en raison des fluctuations de bande passante à court terme , une valeur de changements un peu élevé est enregistré (11), et à partir de l'histogramme précédent, on peut remarquer que le nombre de changements de qualité est diminué de plus de moitié (4) quand on a une révision de flux et cela indique initialement le bon fonctionnement de notre solution.

4.3.2.2 Scenario 2:

Dans ce scénario, nous essayons de démontrer que notre solution réussira à remplacer presque tous les segments indésirables. On programme donc 3 changements de qualité toutes les 20 secondes et nous exécutons notre lecteur avec les paramètres initiaux montrés dans le tableau suivant :

Paramètre	NSR	SR	SP
Valeur	6	30%	3

Tableau 4. 4 paramètres initiaux du 2^{er}scénario.

Pour les 2 cas : **avec et sans révision de flux**, nous calculons le nombre de changements de qualité pendant la vidéo, la bande passante moyenne consommée, le débit binaire moyen de la qualité vidéo, le nombre total des segments téléchargés (les originaux + les remplaçants), et la consommation du CPU.

Nous appliquons notre test sur les 3 vidéos **Bigbunny, Envivio – redbull car, Envivio – monstersshow** et les résultats obtenus sont les suivants :

Vidéo	<i>Big bunny</i>		<i>Envivio – redbull Car</i>		<i>Envivio – monsters show</i>	
	Non	Oui	Non	Oui	Non	Oui
Révision						
moyenne du débit binaire						

mesuré (Kbps)	1837.00	1717.14	1723.30	1656.14	1678.57	1614.53
Moyenne du débit binaire de la qualité du segment sélectionné (kbps)	1243.55	1613.60	989.99	1128.48	1142.37	1315.59
Nombre de changements de qualité	84	02	25	05	27	03
Nombre de segments téléchargés	156	215	65	80	97	115
Consommation CPU(%)	11	9-34	11	10-37	11	10-33

Tableau 4. 5. Résultats du 2^{er} scénario.

Cette courbe ci-dessus présente la bande passante consommée mesurée après le téléchargement de chaque segment dans les 2 cas.

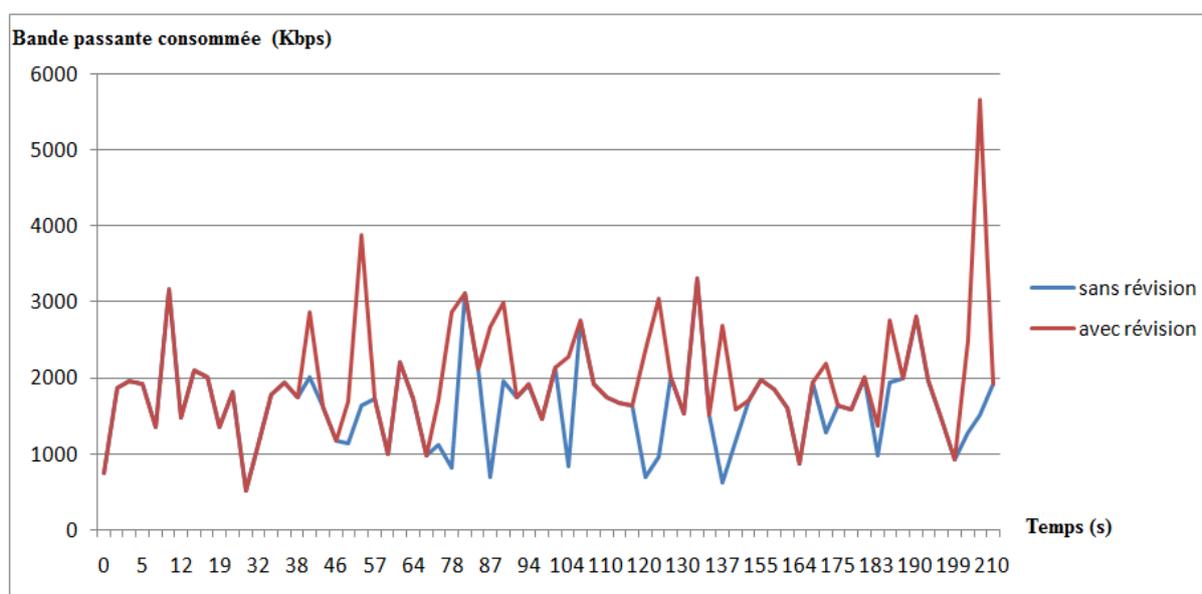


Figure 4. 8. Comparaison de la consommation de données.

Discussion :

- à partir des résultats du tableau on constate que, la moyenne de la bande passante mesurée en cas de révision est différent que sa valeur avec la révision et ceci est démontré par la différence entre le nombre de segments et la variation du temps de récupération de chaque segment.
- selon la courbe, nous remarquons que la consommation de la bande passante est augmentée à certains moments à cause de téléchargement des segments du remplacement. Et d'après le résultat de la 5^{ème} ligne, notre solution a pu diminuer de 80 à 97% des changements de qualité, ce qui démontre la valeur élevée des débits moyens de la qualité des segments sélectionnés en cas de révision.
- Quant à la consommation du CPU, elle est considérablement augmentée du fait du lancement d'un thread supplémentaire pour gérer le téléchargement des segments de remplacement.

4.3.2.3 Scénario 3 :

Dans le 3^{ème} scénario, nous voyons le taux de réussite de la révision si nous exécutons une autre application de streaming qui peut rivaliser avec notre lecteur.

Donc nous lançons l'application de VLC avec notre lecteur et nous enregistrons les résultats suivants :

<i>Big bunny</i>	<i>sans vlc</i>	<i>Avec vlc</i>
débit moyen (Kbps)	1894.81	1095.10
Nombre de changements de qualité (sans révision)	39	47
Nombre de changements de qualité (avec révision)	03	14
Nombre de segments téléchargés	120	117

Tableau 4. 6. Résultat du 3^{ème} scénario.

Avec :

Nombre de segments téléchargés = segments originaux téléchargés + segments de Remplacement téléchargés.

Et le temps nécessaire pour remplacer chaque segment est indiqué dans l'histogramme ci-dessous :

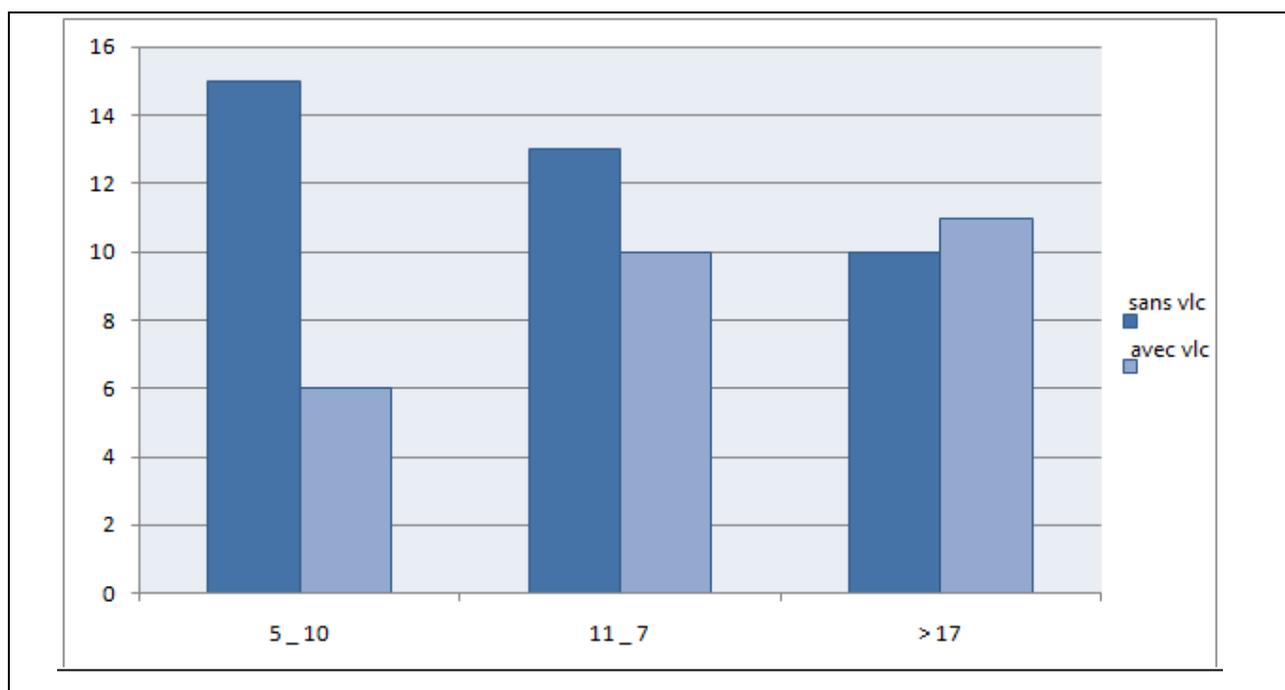


Figure 4. 9. Temps nécessaire pour remplacer les segments.

Discussion :

Du tableau (4.5) , nous remarquons que la moyenne de la bande passante mesurée est plus élevée lorsque nous n'avons que notre programme en cours d'exécution et que le changement de qualité est diminué de 92,3 %. Cependant il est plus faible lors de l'exécution des deux applications et il provoque une diminution de la qualité de la vidéo, mais il diminue toujours le changement de qualité avec une bonne ration qui est égale à 70,2%.

L'histogramme de la figure 4.9 montre qu'en l'absence de VLC, le temps nécessaire au remplacement des segments est beaucoup plus variable dans l'intervalle [5s-10s] et moins dans l'intervalle [11s-17s] et plus faible entre les valeurs supérieures à 17s. Et ce temps enregistré représenté un bon résultat pour le remplacement d'un segment déjà téléchargé existé dans une structure critique. Mais dans le second cas avec VLC, il varie beaucoup plus

dans l'intervalle [11s-17s] et entre les valeurs supérieures à 17s. Mais malgré tout, le processus de changement reste néanmoins réussi.

4.3.2.4 Scénario 4

Dans ce scénario nous essayons de découvrir l'effet de nos paramètres initiaux choisis. Donc nous avons ce que suit :

- **Les conditions initiaux testés** : la condition du débuter la révision lorsqu'on a le tampon plein à 30 % et la condition de débuter la vérification de la 4ème segment.
- Un segment est retiré chaque 5s pour la lecture.
- **temps_LT** : Temps restant à la lecture après le téléchargement du segment (s).
- **temps_LR** : Temps restant à la lecture après remplacements du segment (s).

Nous avons testé notre solution dans des 3situations : limiter la bande passante et ne pas spécifier les paramètres initiaux, ne limiter pas la bande passante et ne pas spécifier les paramètres initiaux, et enfin ne pas limiter la bande passante mais spécifier les paramètres initiaux.

Nous limitons la bande passante en utilisant une application appelée "trickle" sous linux. L'installation et l'utilisation de cette application depuis le terminal est comme suit :

- ✓ Sudo apt-get install trickle.
- ✓ trickle -d (download limit in KB/s) application.

Et nous spécifions la limite de la bande passante pour notre application comme suit :

- ✓ trickle -s -d 65 ./qt-sampleplayer.

I. (+) limitation de la bande passante (-) spécification des paramètres initiaux :

Nous avons obtenons les résultats suivants :

Changements de qualité des segments planifiés	Segments de remplacements Téléchargés	Segments remplacés	Segments de remplacements téléchargés gaspillés

41	28	13	15
----	----	----	----

Tableau 4. 7.Résultats du 1^{er} cas du 4^{ème} scénario.

Et les informations sur l'opération de remplacement de ces 28 segments sont présentées dans le tableau suivant :

Nombre de Segment	Temps_LT	Échec du remplacement	Temps_LR
23	10	Non	5
25	5s	Oui	/
30	15	Oui	/
32	20	Oui	/
35	25	Non	10
39	15	Non	0
40	20	Non	5
43	15	Oui	/
46	20	Non	10
49	20	Oui	/
52	20	Non	10
54	25	Oui	/
56	25	Oui	/
59	25	Non	5s
62	20	Oui	/
65	25	Non	5
68	20	Oui	/
70	20	Non	0
74	10	Oui	/
77	15	Non	5

79	10	Oui	/
81	15	Oui	/
86	20	Oui	/
88	20	Oui	5
91	20	Oui	/
93	15	Non	0
99	15	Oui	/
106	10	Non	5

Tableau 4. 8. Les segments du remplacement téléchargé.

II. (-) limitation de la bande passante (-) spécification les paramètres initiaux :

Nous avons obtenons les résultats suivants :

Changements de qualité des segments planifiés	Segments de remplacements Téléchargés	Segments remplacés	Segments de remplacements téléchargés gaspillés
37	37	36	2

Tableau 4. 9. Résultats du 2^{ème} cas du 4^{ème} scénario

3. (-) limitation de la bande passante (+) spécification les paramètres initiaux :

Nous spécifions les paramètres suivant :

- Longueur du tampon : 20% .
- Révision du 4^{ème} segment.

Et nous avons acquis les résultats suivants :

Changements de qualité des segments planifiés	Segments de remplacement Téléchargés	Segments remplacés	Segments de remplacement téléchargés gaspillés
26	26	26	0

Tableau 4. 10. Résultats du 3^{ème} cas du 4^{ème} scénario

Résultats :

- Dans le cas de limitation de la bande passante, nous avons ce que suit :
 - La longueur du tampon varie entre 0% et 20%.
 - Changements de qualité pendant la vidéo sans révision est : 73.
 - Avec la révision, nous diminuons juste 35,6 % des changements de la qualité (47 changements).
 - Le temps moyens des temps_LR des segments qui n'ont pas pu être remplacés est égal à 17 s.
- Dans le 2^{ème} cas, nous avons ce que suit :
 - La longueur du tampon n'est pas limitée.
 - Changements de qualité pendant la vidéo sans révision est : 72.
 - Avec la révision, nous diminuons 90,4 % des changements de la qualité (7 changements).
 - Temps_LT de 2 segments qu'ont échoués à remplacer est égale à 10s.
 - Temps_LT des autres segments remplacés est =>20s.
- Dans le 3^{ème} cas, nous avons un résultat similaire au résultat du 2^{ème}cas avec aucun segment gaspillé.

Discussion :

- Depuis le résultat du moyen des **temps_LR** des segments qui n'ont pas pu être remplacés dans le 1^{er} cas, nous concluons que la révision doit être appliquée du 4^{ème} segment.

- Dans le 2^{ème} et 3^{ème} cas, la révision était effective dans laquelle les segments téléchargés sont encore loin de la lecture et ils ont suffisamment de temps pour être remplacés avant leur tourne de lecture car le tampon est alimenté à un taux plus élevé qu'il n'est lu.

- Dans le 3^{ème} cas, attendre que le buffer soit plié à plus de 20% et démarrer la révision du 4^{ème} segment n'a pas d'effet considérable sur le résultat de la révision

- Qu'il s'agisse de 73 ou 47 changements de qualité, il est toujours ennuyeux pour les utilisateurs, et il provoque un gaspillage de ressources sans atteindre le résultat souhaité.

-En fonction des résultats dans le 1^{er}, 2^{ème} et 3^{ème} cas, la révision du tampon est efficace lorsque nous débuter la révision à partir de 4^{ème} segments et lorsque nous avons un tampon plein a plus de 20 %.

4.3.2.5 Scenario 5

Dans ce scénario, nous proposons de faire une mesure subjective pour tester la perception de la qualité vidéo par les utilisateurs. La mesure utilisée est donc la note moyenne d'opinion «MOS».

Score moyen d'opinion est une mesure utilisée dans le domaine de la qualité de l'expérience et de l'ingénierie des télécommunications, ce qui représente la qualité globale d'un stimulus ou d'un système. [29]

Nous amenons 12 clients en leur demandant de regarder la vidéo qui est encodée en différentes qualités, une fois via un lecteur « Dash JavaScript » [12] et une autre avec notre lecteur. Ensuite, nous évaluons leurs expériences et calculons le score MOS attribué. Les résultats obtenus sont représentés dans l'histogramme suivant :

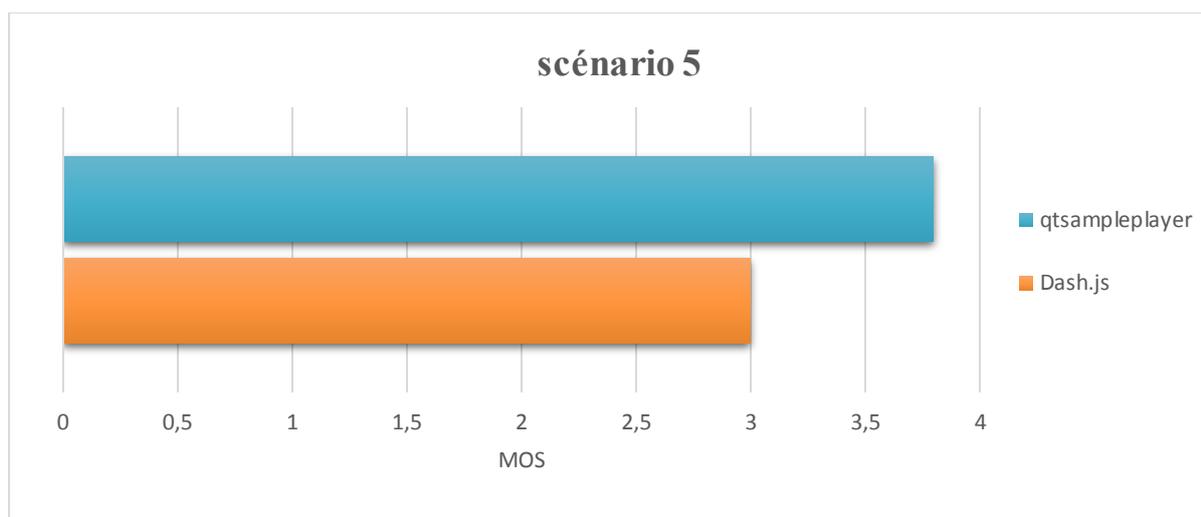


Figure 4. 10. Résultats de 5^{ème} scénario.

Discussion :

A partir de cet histogramme qui représente la note moyenne d'opinion « mos», nous remarquons que lors des tests avec le lecteur DASH.js la moyenne des notes donnée par les clients était 2.9 (étiquette : pauvre), qui est plus faible que la moyenne obtenue dans le deuxième test (notre solution) qui est égal à 3.8 (étiquette : juste), qui est égale presque à la quatrième note de MOS (étiquette : bien). Donc, le résultat obtenu dans notre solution est le meilleur par rapport à l'autre solution. Selon les téléspectateurs, ils ne se soucient pas beaucoup de la qualité de la vidéo car il est important pour eux de ne pas couper la vidéo et surtout d'obtenir une qualité aussi stable que possible parce que les changements de qualité et spécialement de meilleur au pire étaient une nuisance pour eux.

4.4 Conclusion

Nous avons détaillé au cours de ce chapitre l'environnement d'implémentation de notre solution, nous avons parlé des outils d'implémentation en montrant la librairie Libdash, le lecteur QtSAMPLEplayer, puis nous avons discuté sur les différents résultats de notre solution.

Conclusion générale

L'informatique de demain, telle que nous la percevons aujourd'hui, sera distribuée et multimédia. Dans cette optique, le streaming des données audiovisuelles apparaît comme un champ d'application particulièrement prometteur, à condition de disposer des matériels et logiciels susceptibles de répondre aux contraintes très sévères imposées par ce type d'applications tel que conditions d'exécution temps réel avec la meilleure qualité du contenu disponible.

Notre projet a pratiquement atteint ses objectifs initiaux, et l'objectif était d'améliorer la qualité de l'expérience pour les services de streaming vidéo adaptatifs. Mais avant de commencer notre solution, nous avons fait quelques recherches sur ce thème pour nous bien aider à mener le travail et à comprendre le domaine de notre travail. Nous avons commencé avec mentionner des notions de base sur le streaming.

Nous nous sommes intéressés par la suite par la qualité de service et qualité d'expérience donc dans l'état de l'art on a déterminé différentes approches connues pour améliorer ses derniers dans les services de vidéo streaming et on a cité, critiqué et comparé des travaux existants liés à notre travail demandé.

Après avoir cueillis suffisamment d'informations, nous avons proposé une solution qui a opéré sur le buffer des media afin de minimiser les changements de qualité et par conséquence améliorer la qualité de l'expérience de l'utilisateur. Sa pertinence a été validée par implémentation et tests dans le dernier chapitre, les résultats ont montré l'efficacité de notre solution par rapport aux solutions existantes.

Nous considérons que nous avons atteint l'objectif déclaré et supposons que notre approche a répondu à nos attentes.

Nous concluons que la plupart des utilisateurs se soucient de la qualité de la vidéo et en particulier du nombre de changements lors de la visualisation à cause de temps.

Notre solution a été testée par rapport à diverses métriques mais malheureusement nous n'avons pas eu la possibilité de tester la consommation d'énergie en raison de contraintes de matériels.

Pour la suite de notre travail de recherche, la réalisation de plus des tests est un objectif majeur pour nous. D'une autre part, nous envisageons d'améliorer notre stratégie en redéfinissant par exemple les paramètres utilisés.

En outre nous souhaitons que ce travail soit utile et sera appuyé par des autres chercheurs.

BIBLIOGRAPHIE

- [1] SCHULZRINNE, Henning, et al, 2003. “*RTP: A transport protocol for real-time applications*”. Network Working Group .3550. p.89. DOI : 10.17487/RFC3550
- [2] SUH, Dongeun, JANG, Insun, PACK, Sangheon, 2015."QoE-enhanced Adaptation Algorithm over DASH for Multimedia Streaming".Conferencepaper.
- [3] ZHAO, Shuai, LI, Zhu, MEDHI, Deep, 2017.”Study of User QoE Improvement for DynamicAdaptive Streaming over HTTP (MPEG-DASH)”.conference paper.
- [4] ZHANG, Weizhan , HE, Hao, YE, Shuyan, WANG, Zhiwen, ZHENG, Qinghua, 2018. "Enhancing QoE for Mobile Users by Environment-Aware HTTP Adaptive Streaming".Sensors. Vol. 18, No.11, p. 3645.
- [5] LUIGI,Atzori,CHEN,ChangWen.“*Utilisation efficace des serveurs Web en tant que serveurs vidéo pour des applications de vidéo à la demande* ”.
- [6] YE, Zakaria, 2017. “*Analyse de Performance des Services de Vidéo Streaming Adaptatif dans les Réseaux Mobiles*”. Thèse de Doctorat. Informatique. Avignon : Université d’Avignon et des Pays de Vaucluse.
- [7] FELLAG,Cherba , KECHAIRI, Mohamed, 2019 . ”*Utilisation du protocole MPLS pour l’amélioration des paramètres de la QoS d’une chaine de transmission en vidéo streaming*”. Mémoire de master : Réseaux et Télécommunications. telemcen : Université Abou BekrBelkaïd.
- [8] XIAOYU, Ma, JIXUN, Gao, 2012. “*Comparisonand Analysis ofthe Streaming Media Transport Protocol inthe Transmission System* ”. In: International Conference on Education Technology and Computer. 5p.
- [9] Cobalt digitalvInc, 2016. “*IP Video Transport Protocols – Knowing What to Use When and Why*”. Cobalt design white paper.
- [10] NAGASHIMA, Tatsuya, KANAI, Kenji, KATTO, Jiro, 2017. “*QoS and QoE Evaluations of 2K and 4K DASH Contents Distributions*”. IEEE International Symposium on Multimedia. DOI 10.1109/ISM.2017.90.

- [11] European Telecommunications Standard Institute (ETSI), 2010. Universal Mobile Telecommunication System (UMTS); LTE; "Transparent End-to-End Packet Switched Streaming Service (PSS): Progressive download and dynamic adaptive streaming over HTTP (3GP-DASH)". Sophia-Antipolis Cedex, France.
- [12] ISO/IEC, 2012. "Technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media Presentation Description and Segment Formats".
- [13] SEDRAT, Saida, 2012. "Gestion de la Qualité de Service des flux Streaming dans les Réseaux 802.11". Mémoire de Magister. Ingénierie des Systèmes Informatique. Batna : Université El Hadj Lakhdar.
- [14] DUBIN, Ran, HADAR, Ofer, DIVIR Amit, 2013. "The Effect of Client Buffer and MBR Consideration on DASH Adaptation Logic". IEEE Wireless Communications and Networking Conference (WCNC): NETWORKS.
- [15] European Horizon, 2017. Content adaptation using DASH streaming services. projet de recherché Easytv.
- [16] DAJIE, Wang, 2016. Dynamic Adaptive Streaming over HTTP. thèse de master. Computer Science. Helsinki : Université d'Helsinki.
- [17] BOURAQIA, Khadija, SABIR, Essaid, SADIK, Mohamed, LADID, Latif, 2020. "Quality of Experience for Streaming Services: Measurements, Challenges and Insights". IEEE Access. Vol. 8, No. , p. 13341-13361.
- [18] FAREED GHANI, SUFIUH, Rana, AJRASH, Amal, 2018. "Quality of Experience Metric of Streaming Video: A survey." Iraqi Journal of Science, Vol. 59, No.3B, p. 1531-1537.
- [19] Github, 2013. Fun with Container Formats – Part 2 [en ligne]. Disponible sur : <https://github.com/bitmovin/libdash> [consulté le 12 juillet 2020].
- [20] GRAVEY, Annie, 2013. La dissémination et la valorisation .Rapport . Projet VIPEER. Brest: Telecom Bretagne.

[21] GUILLEN, Luis, ZUMI, Satoru I, ABE, Toru, SUGNUM, Takuo, 2019. “*SAND/3: SDN-Assisted Novel QoE Control Method for Dynamic Adaptive Streaming over HTTP*”. *Electronics*, Vol. 8, No. 8.

[22] HISAMATSU, Hiroyuki, HASEGAWA, Go, MURATA, Masayuki, 2011. Eighth International Conference on Information Technology: New Generations, ITNG. 11-13 April 2011, Las Vegas, Nevada, USA

[23] Index of /129021/dash. Disponible sur : <https://dash.akamaized.net/> [consulté le 1er septembre 2020].

[24] DASH INDUSTRY FORUM. Reference Client 3.1.3 [en ligne]. Disponible sur : <http://reference.dashif.org/dash.js/nightly/samples/dash-if-reference-player/index.html> [consulté le 1er septembre 2020].

[25] KIM, Sangook, YUN, Dooyeol, et, CHUN, Kwangsue, 2016, “*Video quality adaptation scheme for improving QoE in HTTP adaptive streaming*”. 2016 International Conference on Information Networking (ICOIN). Kota Kinabalu. pp. 201-205. Doi 10.1109/ICOIN.2016.7427114.

[26] LAHBABI, Youssef, IBN ELHAJ, El Hassan, HAMMOUCH, Ahmed, 2014. , “*Quality adaptation using Scalable Video Coding (SVC) in Peer-to-Peer (P2P) Video-on-Demand (VoD) streaming* ”. 2014 International Conference on Multimedia Computing and Systems (ICMCS), Marrakech. pp. 1140-1146. DOI : 10.1109/ICMCS.2014.6911236.

[27] BRUNEAU-QUEYREIX, Joachim, LACAUD, Mathias, NÉGRU, Daniel, 2017. “*A Hybrid P2P/Multi-Server Quality Adaptive Live-Streaming Solution Enhancing End-User’s QoE*”. ACM MultiMedia, Mountain View, United States.

[28] DIALLO, Mamadou Tourad, 2015. “Qualité d’Expérience et Adaptation de services vidéo”. Thèse de doctorat. Informatique et Télécommunication. Sud Paris : Ecole Doctorale EDITE.

[29] Mean opinion score. Disponible sur : https://fr.qwe.wiki/wiki/Mean_opinion_score [consulté le 2 septembre 2020].

- [30] DAGIUKLAS, Tasos et al, 2016. “*Multimedia Quality of Experience (QoE): Status and Future Requirements*”.
- [31] REUBAN, GnanaAsir, KiSHORE, Kumar C, PRAVEEN, Kumar Reddy.M, 2014. “*MPEG-DASH Enhanced Multimedia Streaming*”. International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, No. 3.
- [32] APOSTOLOPOULOS John G, TAN, Wai- tian, WEE, Susie J, 2002. "*Video Streaming: Concepts, Algorithms, and Systems*". Technical Report *HPL-2002-260*, HP Laboratoires.
- [33] [33] DOUGA, Yacine, 2016. “*Amélioration de la performance de TCP dans les réseaux mobiles ad hoc*”. These de doctorat. Oran: Université Ahmed Ben Bella.
- [34] Bitmovin, 2019. Fun with Container Formats – Part 2[en ligne]. Disponible sur : <https://bitmovin.com/fun-with-container-formats-2/> [consulté le 12 juin 2020].
- [35] Techopedia, 2017. Techopediaexplains Ubuntu[en ligne]. Disponible sur:<https://www.techopedia.com/definition/3307/ubuntu>[consulté le 20 AOUT 2020].
- [36] Techtarget, 2009. Search data center [en ligne]. Disponible sur : <https://searchdatacenter.techtarget.com/definition/Ubuntu>[consulté le 20 AOUT 2020].
- [37] Christopher, Mueller, et al, 2013. “*DEMO PAPER: LIBDASH - AN OPEN SOURCE SOFTWARE LIBRARY FOR THE MPEG-DASH STANDARD*”. IEEE conferences. DOI: [10.1109/ICMEW.2013.6618220](https://doi.org/10.1109/ICMEW.2013.6618220).
- [38] BOUDJAADAR, Amina, 2010. "*Plateforme basée Agents pour l'aide à la conception et la simulation des réseaux de capteurs sans fil*". Thèse de Magistère. Skikda: Université de Skikda.
- [39] BIERNACKI, Arkadiusz, TUTSCHKU, Kurt, 2014. Performance of HTTP video streaming under different network conditions. Multimedia Tools and Applications. Vol.72, p.1143–1166. DOI [10.1007/s11042-013-1424-x](https://doi.org/10.1007/s11042-013-1424-x).
- [40] BRUNEAU-QUEYREIX, Joachim, LACAUD, Mathias, NEGRU, Daniel, BATALLA, Jordi Mongay, BORCOCI, Eugen, 2017. , "*MS-Stream: A multiple-source adaptive streaming solution enhancing consumer's perceived quality*". 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas. pp. 427-434. Doi: [10.1109/CCNC.2017.7983147](https://doi.org/10.1109/CCNC.2017.7983147).