

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ de BLIDA 1
Faculté de Science
Département D'informatique



Mémoire

En vue de l'obtention du diplôme de

MASTER EN INFORMATIQUE

Spécialité : Systèmes Informatiques et Réseaux

Intitulé du mémoire

**Amélioration de la qualité d'expérience des services de
vidéo streaming adaptatif dans une architecture MEC**

Présenté par :

M^{lle} KELLECHE Soumia.
M^{lle} AHMED M'barek Hafsa.

Encadré par :

Dr. DOUGA Yassine

Devant le jury :

Mme H.YKHLEF

Présidente

Mr M.BENYAHIA

Examineur

Année universitaire 2019/2020

Remerciements

On remercie Dieu tout Puissant de nous avoir donné la force, la santé, le courage et la patience de pouvoir accomplir ce travail qui est pour nous le point de départ d'une merveilleuse aventure, celle de la recherche, source de remise en cause permanent et de perfectionnement perpétuelle.

Un grand merci à toutes nos familles surtout nos parents pour leur encouragement et leur suivi avec patience le déroulement de notre projet.

*Nous tenons à remercier également notre promoteur **M. DOUGA Yassine** d'avoir accepté de nous guider, pour ses orientations, son soutien tout au long de ce travail.*

*Nous tenons à exprimer nos vifs remerciement à tous ceux qui
on
Collaborer avec leurs idées, leurs expériences ont participé de
près ou de loin à la réalisation de ce travail.
Enfin, nous tenons aussi à remercier également tous les
membres
du jury pour avoir accepté d'évaluer notre travail.*

Dédicaces

Rien n'est aussi beau à offrir que le fruit d'un labeur qu'on dédie du fond du cœur à ceux qu'on aime et qu'on remercie en exprimant la gratitude et la reconnaissance durant toute notre existence

Je dédie ce travail à :

A ma très chère mère, que nulle dédicace ne puisse exprimer mes sincères sentiments, pour sa patience illimitée, s'encouragement continu, son aide, en témoignage de mon profond amour et respect pour sa grande sacrifices.

*A mon cher frère **Sief Eddine** et mes chères cousines **Khadidja, Meriem** pour leur grand amour et leur soutien qu'elles trouvent ici l'expression de ma haute gratitude.*

Mes chers amis qui sans leur encouragement ce travail n'aura jamais vu le jour.

*Je tiens également à remercier monsieur **Yahia** et monsieur **Mohamed Seddik** pour leurs échanges et discussions fort intéressantes*

*Et à toute la famille « **Kelleche** » et la famille « **Lakder** » et à tous ceux que j'aime.*

Soumia

Dédicaces

Je dédie ce travail

À mes chers parents, aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

A mes chers et adorable frères et sœurs.

A ma nièce «Ahlem ».

A mes meilleures amies « Raouf » « Nesrine » « Sihem » « Meriem » et « Khaoula » qui n'ont cessé jamais de me soutenir.

A tous ceux que j'aime et qui m'aime.

Hafsa.

ملخص

إن بث الفيديو التكيفي عبر HTTP والذي يعرف باسم بث HTTP التكيفية (HAS), هو تقنية بث فيديو منتشرة على نطاق واسع عبر الإنترنت لخدمات الفيديو (Live) والفيديو حسب طلب (VOD). يشمل ذلك تقسيم الفيديو الأصلي المخزن على خادم HTTP إلى مقاطع صغيرة (عادة مع نفس وقت التشغيل). عندما يشارك العديد من العملاء نفس الارتباط سيكون هناك اختناق ومنافسة لعرض النطاق الترددي. وفي هذه الحالة فإن جودة التجربة (QOE) وجودة الخدمة (QOS) لشبكة الوصول كثيرا ما تتدهور.

في هذه الرسالة، قمنا بإجراء دراسات حول الأساليب التي تركز على تحسين جودة الخبرة، وقد اخترنا النهج الأكثر فائدة والأحدث، وقد قمنا بإجراء دراسة تفصيلية حول هذا الموضوع، وآخر هذه الدراسات تتعلق بتقييم الأداء، حيث تمكنا من سد بعض الثغرات وإبراز مشكلة البطارية والازدحام بهدف اقتراح التحسينات.

الكلمات المفتاحية:

جودة التجربة (QOE)، تدفق الفيديو، الحوسبة المتنقلة (MEC)، محطة المستخدم، الازدحام.

Résumé

Le streaming vidéo adaptatif sur HTTP, généralement appelé HTTP Adaptive Streaming (HAS), est une technique de streaming vidéo largement déployée sur Internet pour les services de vidéo en temps réel (Live) et de vidéo à la demande (VoD). Cela inclut la division de la vidéo originale stockée sur le serveur HTTP en petits segments (généralement avec le même temps de lecture). Lorsque plusieurs clients partagent le même lien, il y aura un goulot d'étranglement (bottleneck) et une concurrence pour la bande passante. Dans ce cas, on signale fréquemment une dégradation de la qualité d'expérience (QoE) des utilisateurs et de la qualité de service (QoS) du réseau d'accès [24].

Dans ce présent mémoire nous avons fait des études sur des approches qui mettent l'accent sur l'amélioration de la qualité d'expérience, parmi ces approches nous avons choisi celle qui est la plus avantageuse et la plus récente, sur laquelle nous avons mené une étude détaillée, cette dernière porte sur l'évaluation de la performance, où nous avons pu extraire à la sortie quelques lacunes et mettre en valeur le problème de batterie et de congestion en vue de suggérer des améliorations.

Mots clés: Qualité d'expérience (QoE), Streaming vidéo, Mobile edge computing (MEC), Terminal d'utilisateur, Congestion.

Abstract

Adaptive video streaming over HTTP, commonly referred to as HTTP Adaptive Streaming (HAS), is a widely deployed video streaming technique over the Internet for real-time video (Live) and video-on-demand (VoD) services. This includes dividing the original video stored on the HTTP server into small segments (usually with the same playback time). When multiple customers share the same link, there will be bottleneck and bandwidth competition. In this case, the quality of experience (QoE) of the users and the quality of service (QoS) of the access network is frequently degraded.

In this brief we've done studies on approaches that focus on improving the quality of experience, and we've chosen the most advantageous and most recent approach, which we've done a detailed study on. This last one is about performance evaluation, where we were able to pull out some gaps and highlight the battery and congestion problem with a view to suggesting improvements.

Keywords: Quality of Experience (QoE), Video Streaming, Mobile Edge Computing (MEC), User Terminal, Congestion.

Table de matières

<i>Remerciements</i>	i
<i>Dédicaces</i>	ii
ملخص	iv
Résumé	v
Table de matières	vii
Liste des figures	x
Liste des tableaux	xi
LISTE DES ACRONYMES	xii
INTRODCTION GENERALE	1
Chapitre I : Notions de base	4
I.1 Introduction.....	4
I.2 Le Streaming.....	4
I.2.1 Définition.....	4
I.2.2 Les protocoles de streaming	5
I.2.3 Classification des technologies de streaming.....	5
I.2.4 Evolution du streaming multimédia	6
I.3 La qualité de service	7
I.3.1 Définition	7
I.3.2 Niveaux de service	8
I.3.3 Critères de qualité de service	8
I.4 Qualité de l'expérience	9
I.4.1 Définition	9
I.4.2 Les approches de la QOE	9
I.4.3 Les métriques de QOE	10
I.5 Relations entre QoS et QoE	11
I.6 Mobile Edge Computing (MEC)	11
I.6.1 Définition	12
I.6.2 Les caractéristiques de la technologie MEC	12
I.6.3 Architecture MEC	13
I.7 Conclusion	14
Chapitre II :Etat de l'art de vidéo streaming adaptatif	16
II.1 Introduction.....	16

II.2	Approches récentes pour l'amélioration du streaming adaptatif basées sur l'utilisateur	16
II.2.1	Approche 1 : Streaming adaptatif dynamique sur HTTP (DASH)	16
II.2.2	Approche 2 : TCP et l'amélioration des services de vidéo streaming adaptatif	19
II.2.3	Approche 3 : Amélioration de QOE des services de vidéo streaming adaptatif dans une architecture MEC	23
II.3	Etude comparative	28
II.4	Conclusion	29
Chapitre III : Etude des performances de l'approche AVSCU		31
III.1	Introduction.....	31
III.2	L'environnement de l'émulation	31
III.2.1	La topologie d'émulation.....	31
III.3	Les scénarios du test	32
III.3.1	Scenario 1 : contrôle de la congestion.....	32
III.3.2	Scenario 2 : nombre des coupures	34
III.3.3	Scenario 3 : consommation de la batterie.....	36
III.3.4	Scenario 4 : consommation du CPU.....	38
III.3.5	Scenario 5 : consommation du forfait internet	39
III.4	Récapitulatif des scenarios.....	41
III.5	Conclusion	42
Chapitre IV : Conception d'un mécanisme d'amélioration de la QoE dans les services de vidéo streaming adaptatif		44
IV.1	Introduction.....	44
IV.2	Mécanisme du fonctionnement	44
IV.3	Les étapes de la solution proposée.....	45
IV.4	Conclusion	49
Chapitre V : Implémentation et étude expérimentale		51
V.1	Introduction.....	51
V.2	Outils d'implémentation	51
V.3	Les Scénarios du test.....	54
V.3.1	Scénario 1	55
V.3.2	Scénario 2	56
V.3.3	Scénario 3	57

V.4 Conclusion	58
CONCLUSION GENERALE	60
Bibliographie	62
Annexe	65

Liste des figures

<i>Figure I.1: Streaming multimédia.</i>	4
<i>Figure I.2: Relation entre QoS et QoE.</i>	11
<i>Figure I.3: Modèle de service a 3 niveaux .</i>	14
<i>Figure II.1: L'architecteur de l'approche .</i>	17
<i>Figure II.2: Diagramme de flux pour les algorithmes d'adaptation pour DASH.</i>	18
<i>Figure II.3: Architecture de l'approche</i>	23
<i>Figure III.1: Résultat de scénario 1.</i>	33
<i>Figure III.2: Résultat de scénario 2.</i>	35
<i>Figure III.3: Résultat de scénario 3.</i>	37
<i>Figure III.4: L'exécution de la commande « htop ».</i>	38
<i>Figure III.5: Résultat de scénario 4.</i>	39
<i>Figure III.6: Résultat de scénario 5.</i>	40
<i>Figure V.1: Topologie du streaming.</i>	51
<i>Figure V.2: La page du site web.</i>	53
<i>Figure V.3: Résultat de scénario 1.</i>	55
<i>Figure V.4: Résultat de scénario 2.</i>	56
<i>Figure V.5: Résultat de scénario 3.</i>	57

Liste des tableaux

<i>Tableau I.1: Classification des technologies de streaming .</i>	6
<i>Tableau I.2: La note moyenne d'opinion MOS.</i>	10
<i>Tableau II.1: Attribution de qualité.</i>	24
<i>Tableau II.2: Consommation de bande passante.</i>	25
<i>Tableau II.3: MOS selon les états.</i>	26
<i>Tableau II.4: Comparaison des approches.</i>	29
<i>Tableau III.1: Caractéristiques de chaque PCs.</i>	32
<i>Tableau III.2: Paramètres des terminaux utilisés dans l'émulation.</i>	32
<i>Tableau IV.1: Valeur de FB.</i>	45
<i>Tableau IV.2: Valeur de FR.</i>	46
<i>Tableau IV.3: Valeur de FT.</i>	46
<i>Tableau IV.4 : L'attribution de la qualité.</i>	46
<i>Tableau IV.5 : L'attribution de la qualité.</i>	47
<i>Tableau IV.6: La consommation de la bande passante .</i>	47

LISTE DES ACRONYMES

3GPP: 3rd Generation Partnership Project.

5G: 5th Generation of Wireless Communication Technology.

AVSCU: Adaptation de Vidéo Streaming Centralisé basé sur l'Utilisateur.

BD: La Batterie Disponible.

CDN: Content Delivery Network.

CP: Central Processing Unit.

DASH: Dynamic Adaptive Streaming over HTTP.

EnodeB: Evolved Node B.

FB : Le Facteur de Batterie.

FM : Le Facteur de MOS.

FR : Le Facteur de Résolution.

FS : Le Facteur de la taille d'écran.

FU : Le Facteur Utilisateur.

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.

MITMProxy: Man In The Middle Proxy.

MOS: Mean Opinion Score.

MPD: Media Presentation Description.

MPEG: Moving Picture Experts Group.

PBT: Bande Passante Totale.

PBU: Bande Passante des utilisateurs.

RC: Risque de Congestion.

PDM: Processus de Décision de Markov.

QoE: Quality of experience.

QoS: Quality of service.

RAM: Random Access Memory.

RAN: Radio Access Network.

RSVP: Ressource ReSerVation Protocol.

RT: La Résolution du Terminal.

RTCP: Real-time Transport Control Protocol.

RTP: Real-time Transport Protocol.

RTSP: Real Time Streaming Protocol.

TCP: Transport Control Protocol.

TE: La taille de l'écran.

UDP: User Datagram Protocol.

UF: User Factor.

Wifi: Wireless Fidelity.

XML: Extensible Markup Language.

INTRODUCTION GENERALE

Les plateformes de vidéo en streaming font désormais le bonheur de nos moments de divertissement vidéo. Ainsi, ce mode de consommation de contenus audiovisuels, s'ancre dans les habitudes de tout à chacun. En effet, elles sont bien aidées par l'avènement d'internet, des « smart TV » ainsi que des appareils mobiles connectés [25].

On peut dire donc que le trafic de streaming vidéo est de loin le trafic applicatif dominant sur internet d'aujourd'hui [26], L'optimisation de la diffusion vidéo dépend de la capacité à déterminer la qualité de vidéo flux qu'un utilisateur reçoit.

Les utilisateurs des services de streaming sont devenus très exigeant sur la qualité reçue de la vidéo et donc la qualité d'expérience. La qualité d'expérience reflète la satisfaction des utilisateurs vis-à-vis des services qu'ils utilisent [5].

Dans ce cas, adapté à notre projet de recherche le but est d'améliorer la qualité d'expérience des services de vidéo streaming adaptatif dans une architecture MEC (Mobile EdgeComputing).

Dans notre solution proposée, nous avons basé sur deux points importants la consommation énergétique « la consommation de la batterie » et le contrôle de la congestion dans cette architecture.

Le reste de ce document est organisé comme suit :

- Chapitre 1 « Notions de base » : nous présentons dans ce chapitre une généralité sur le streaming par exemple la définition de cette technologie, les différents protocoles... etc. On va parler de la qualité de service et la qualité d'expérience et la relation entre eux. Enfin, on va présenter l'architecture MEC.
- Chapitre 2 « L'état de l'art » : Nous présentons dans ce chapitre quelques approches connues pour améliorer la qualité d'expérience dans les services de vidéo streaming qui se basent sur les paramètres du terminal final, Nous finirons par une étude comparative.
- Chapitre 3 « Etude de performance de l'approche AVSCU » : Ce chapitre est consacré à l'évaluation de l'approche AVSCU et de trouver ses points de forts et ses faiblesses afin de les améliorer.

- Chapitre 4 « Conception d'un mécanisme d'amélioration de la QoE dans les services de vidéo streaming adaptatifs » : Ce chapitre constitue l'approche que nous proposons, on a détaillé la solution en citant son principe, sa description et les algorithmes utilisés pour l'adaptation de vidéo streaming.
- Chapitre 5 « Implémentation et étude expérimentale » : Nous exposerons les différents outils utilisés durant l'implémentation, nous avons proposé quelques scénarios afin d'évaluer la solution proposée en discutant les résultats de chacun.
- Conclusion : Enfin, nous présentons une conclusion générale du travail présenté et nous proposons quelques perspectives pour de futurs travaux de recherche qui s'inscrivent dans la continuité de ce mémoire.

Chapitre I : Notions de base

Chapitre I : Notions de base

I.1 Introduction

Nous utilisons la technologie de streaming en ligne depuis un bon moment maintenant, et c'est même devenu pour certains leurs moyen principal de consommation internet (sur Youtube ou Netflix par exemple).

Ce chapitre vise à présenter d'abord le streaming vidéo, citer les différents protocoles de streaming la classification de ses technologies puis l'évolution du streaming media, ensuite nous allons parler de la qualité de service, la qualité de l'expérience et la relation entre les deux. Enfin, il se termine par décrire l'architecture MEC.

I.2 Le Streaming

I.2.1 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). [1]

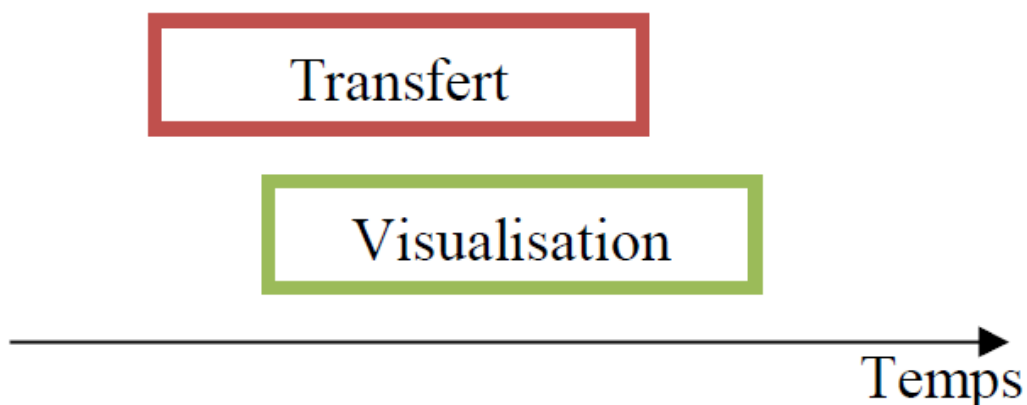


Figure I.1: Streaming multimédia[1].

I.2.2 Les protocoles de streaming

- a. **Real-time Transport Protocol (RTP):** RTP est un protocole de transfert des données en temps réel basé sur IP, qui se situe au niveau de l'application. Il élimine la qualité fournie par TCP (pour gérer les paquets perdus) afin de fournir des applications en temps réel via UDP (plus rapide) [2].
- b. **Real-Time Control Protocol (RTCP):** Le protocole RTCP est basé sur des transmissions périodiques de paquets de contrôle par tous les participants dans la session. C'est un protocole de contrôle des flux RTP, permettant de véhiculer des informations basiques sur les participants d'une session. [3]
- c. **Real Time Streaming Protocol (RTSP) :** est un protocole qui fournit un cadre pour le transfert de données de médias en temps réel au niveau de l'application. Le protocole se concentre sur la connexion et le contrôle des sessions de diffusion de données multiples sur les lignes de synchronisation temporelle pour les médias continus comme la vidéo et l'audio [4].

I.2.3 Classification des technologies de streaming

On peut classer les technologies de streaming selon l'adaptation ou selon le protocole de transport [5], comme montre le tableau suivant :

Selon l'adaptation	Selon le protocole de transport
<p>Streaming adaptatif : il est très utilisé pour améliorer l'expérience d'utilisateur final. Il prend en charge pour améliorer la qualité d'expérience des paramètres tels que : la bande passante disponible, la résolution d'affichage, la puissance du processeur, l'état tampon, etc.</p>	<p>Streaming vidéo basé sur TCP : TCP c'est un protocole très utilisé sur Internet mais généralement qu'il ne convient pas au streaming multimédia et pour cela le protocole (HTTP) a été utilisé sur TCP. Pour le streaming vidéo basé sur HTTP - TCP, il existe essentiellement deux catégories: HTTP progressif téléchargement et diffusion adaptative HTTP (HAS).</p>
<p>Le streaming non adaptatif : contrairement au streaming adaptatif l'utilisateur dispose de multiples qualités et peut choisir manuellement la bonne qualité vidéo qui va avec ces ressources. La vidéo va souffrir d'interruptions si l'utilisateur ne dispose pas suffisamment de bandes passantes, des ressources du processeur ou de la mémoire.</p>	<p>Streaming vidéo basé sur UDP : UDP est très prisé pour les applications temps réel de type audio/vidéo principalement parce qu'on peut tolérer des pertes de paquets. Ce qui n'est pas le cas si on veut transférer par exemple un gros fichier texte entre un émetteur et un récepteur (là, il faut recevoir la copie exacte de ce qui est envoyé).</p>

Tableau I-1: Classification des technologies de streaming [5].

I.2.4 Evolution du streaming multimédia [6]

- a. **Les années 2000**: la grande majorité du trafic Internet était basée sur le protocole HTTP et les réseaux ont été de plus en plus utilisés. Le streaming, avec son assortiment de protocoles propriétaires, basés principalement sur le protocole UDP peu populaire, s'est retrouvé de lui-même à ne plus réussir à répondre à la demande croissante des utilisateurs. C'est en 2007 que la société « Move Networks» introduisit un nouveau concept qui fit grandement évoluer le streaming au niveau international : le streaming adaptatif basé sur HTTP.

- b. **L'arrivée du Streaming adaptatif basé sur http:** la société « Move Networks » décida d'utiliser le protocole HTTP, pour délivrer les contenus multimédias sous forme de petits paquets de données. Cela permit aussi d'utiliser l'application de lecture vidéo pour superviser la vitesse de téléchargement et demander des paquets de qualités variables en s'adaptant automatiquement aux conditions du réseau disponible. il a permis aux vidéos d'être distribuées un peu partout à l'aide des réseaux standards opérateurs et d'être mis en cache pour le passage à l'échelle, tout en éliminant en même temps des problèmes de tampon ennuyeux et des problèmes de connectivité pour les clients.
- c. **L'arrivée des géants du Web:** Microsoft lança sa technologie « Smooth Streaming » en 2008. Apple a emboîté le pas en 2009 avec son protocole propriétaire « HLS (HTTP Live Streaming)» et Adobe rejoint ses collègues en 2010 avec un nouveau protocole nommé « HDS (http Dynamic Streaming) ». Le streaming est devenu très rapidement un outil de choix pour la diffusion en live d'événement majeur (Jeux olympiques de Londres, Wimbledon, Roland-Garros, etc.).
- d. **Le travail de normalisation :** Dès 2009, des efforts ont été faits au sein de la 3GPP pour établir un standard industriel concernant le streaming adaptatif. Très vite, ces travaux de standardisation ont été transmis à des groupes de travail de l'ISO et du comité MPEG. En 2010, les travaux se sont rapidement transformés en « Draft » jusqu'à sa ratification en moins de deux ans. C'est ainsi qu'en avril 2012, une nouvelle norme est née : le streaming adaptatif dynamique sur HTTP, plus connu sous le nom de MPEG_DASH.

I.3 La qualité de service [7]

I.3.1 Définition

Le terme **QoS** (acronyme de « Quality of Service », en français « Qualité de Service ») désigne la capacité à fournir un service (notamment un support de communication) conforme à des exigences en matière de temps de réponse et de bande passante.

I.3.2 Niveaux de service

Le terme « **niveau de service** » (en anglais *Service level*) définit le niveau d'exigence pour la capacité d'un réseau à fournir un service point à point ou de bout en bout avec un trafic donné on distingue 3 niveaux :

- **Meilleur effort** (en anglais *best effort*), ne fournissant aucune différenciation entre plusieurs flux réseaux et ne permettant aucune garantie.
- **Service différencié** (en anglais *differentiated service* ou *soft QoS*), permettant de définir des niveaux de priorité aux différents flux réseau sans toutefois fournir une garantie stricte.
- **Service garanti** (en anglais *guaranteed service* ou *hard QoS*), consistant à réserver des ressources réseau pour certains types de flux. Le principal mécanisme utilisé pour obtenir un tel niveau de service est RSVP (*Resource reSerVation Protocol*, traduisez *Protocole de réservation de ressources*).

I.3.3 Critères de qualité de service

Les principaux critères permettant d'apprécier la qualité de service sont les suivants :

- **Débit** (en anglais *throughput*), il définit le volume maximal d'information (bits) par unité de temps.
- **Gigue** (en anglais *jitter*) : elle représente la fluctuation du signal numérique, dans le temps ou en phase.
- **Latence, délai ou temps de réponse** (en anglais *delay*) : elle caractérise le retard entre l'émission et la réception d'un paquet.
- **Perte de paquet** (en anglais *packetloss*): elle correspond à la non-délivrance d'un paquet de données, la plupart du temps dûe à un encombrement du réseau.
- **Dé-séquencement** (en anglais *desequencing*) : il s'agit d'une modification de l'ordre d'arrivée des paquets.

I.4 Qualité de l'expérience

I.4.1 Définition

Le concept de qualité d'expérience apparaît dans de nombreuses disciplines telles que la psychologie sociale, les sciences cognitives et l'économie. En technologie de l'information, l'évaluation de la qualité d'expérience dépend principalement de la perception de l'utilisateur et de son degré de satisfaction ou d'irritation à l'égard d'un service ou d'une application. [8]

I.4.2 Les approches de la QOE [6]

A. **Approches objectives** : sont basées sur des techniques mathématiques ou/et comparatives qui génèrent des mesures quantitatives de la qualité d'un service proposé à un utilisateur. Ce type d'approche est utile pour la surveillance de la qualité en cours d'utilisation d'un service, pour la conception du réseau des terminaux utilisés, ainsi que dans l'optimisation de la sélection des codecs à utiliser.

Il existe deux types d'approches objectives :

- a. **Les méthodes intrusives** : sont basées sur les signaux, elles sont exactes, mais ne sont pas utilisables pour la supervision du trafic en temps réel. Ceci à cause de la nécessité d'avoir à disposition la source du service comme modèle de référence pour la mesure pour la mesure de la qualité.
 - b. **Les méthodes non intrusives** : elles ne nécessitent pas d'acquérir l'original du service comme référence. Elles peuvent donc être utilisées pour une mesure de la qualité de l'expérience en direct.
- B. **Approche subjective** : sont généralement basées sur des tests d'utilisateurs devant évaluer différentes configurations d'un service. Généralement basés sur la recherche d'un seuil d'acceptabilité général permettant de contenter la majorité des utilisateurs. Le principal objectif de ces études est de définir les différents seuils de qualité pour la diffusion multimédia.
- a. **Score d'opinions moyen** : est une méthode standardisée dans une recommandation de l'ITU-T. Ce type de test est très souvent utilisé, car il est le seul à prendre en considération la subjectivité inhérente aux ressentis individuels de chaque utilisateur.

Le principe de calcul du MOS est basé sur un sondage d'un échantillon supposé représentatif de la population des utilisateurs. Les personnes constituant l'échantillon sont invitées à regarder ou écouter, puis son équivalent codé décodé. À la fin, ils donnent une note sanctionnant la qualité qu'ils ont perçue. La moyenne des notes fournies par la population constitue le MOS [5].

Le tableau ci-dessous montre les notes et leurs significations en fonction de qualité :

Note	Qualité
1	Excellente
2	Bien
3	Juste
4	Pauvre
5	Mal

Tableau I.2: la note moyenne d'opinion MOS [5].

- b. **Défauts du MOS** : Certains défauts lui sont souvent reprochés comme :
- Son coût de mise en place important.
 - Une mise en place et une réalisation très chronophages.
 - Le fait qu'il ne peut pas être utilisé en temps réel.
 - Un manque de répétabilité.

I.4.3 Les métriques de QOE [6]

- Délai initial ou délai de démarrage** : il désigne la durée (mesurée en secondes) entre le moment où un utilisateur lance une session vidéo et le moment où le lecteur multimédia commence à lire les images vidéo.
- Le ratio de mise en mémoire tampon** : est le rapport entre le temps que le lecteur vidéo passe en mode blocage et le temps de lecture (y compris le temps de blocage).
- Taux de mise en mémoire tampon** : c'est la fréquence à laquelle les événements de mise en mémoire tampon se produisent pendant la session vidéo.
- Débit binaire moyen** : est la qualité moyenne de la vidéo diffusée en continu.

- e. **Nombre de changement de débit** : il indique le nombre de fois que la qualité vidéo change pendant la session.

I.5 Relations entre QoS et QoE [6]

Comme présenté par Fiedler un lien peut exister entre le niveau de perception d'un utilisateur et les perturbations engendrées sur la vidéo lors de sa compression ou de sa diffusion. Ceci revient à dire qu'un lien peut exister entre la QoS et la QoE.

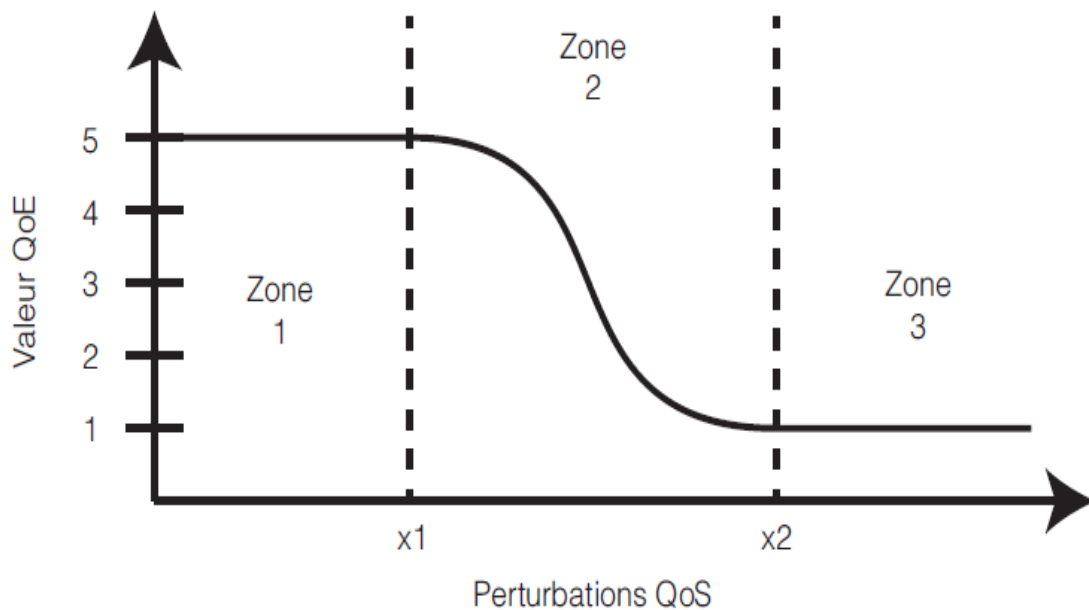


Figure I.2: Relation entre QoS et QoE[6].

Lorsque la perturbation liée à la qualité de service est comprise dans la zone 1, la QoE a une valeur élevée, ce qui signifie que l'expérience utilisateur n'est pas affectée. Le service est donc rendu de manière optimale.

Dès que l'on atteint la zone 2, la qualité de l'expérience commence à décroître en lien des perturbations de la QoS. Dès que l'on atteint la zone numéro trois, les perturbations de QoS perturbent trop l'expérience utilisateur. Il y a donc une probabilité non négligeable qu'il coupe le service en raison de son non satisfaction.

On peut ainsi constater que lorsque les perturbations du QoS augmentent, les métriques de qualité de l'expérience et de qualité de la perception de l'utilisateur décroissent.

I.6 Mobile Edge Computing (MEC) [9]

L'émergence de plusieurs nouvelles applications informatiques, telles que la réalité virtuelle et les environnements intelligents, est devenue possible grâce à la disponibilité d'un vaste ensemble de ressources et de services en nuage. Cependant, les applications sensibles aux délais posent des exigences strictes en matière de délais qui transforment l'euphorie en problème. Le paradigme de l'informatique dans les nuages est incapable de répondre aux exigences de faible latence, de connaissance de la localisation et de soutien à la mobilité.

I.6.1 Définition

Le Mobile Edge Computing (MEC) est une technologie émergente où les services d'informatique en nuage sont étendus à la périphérie des réseaux en exploitant les stations de base mobiles. En tant que technologie de pointe, elle peut être appliquée aux technologies mobiles, sans fil et filaires des scénarios, en utilisant des plateformes logicielles et matérielles, situées à le bord du réseau à proximité des utilisateurs finaux. La MEC fournit l'intégration transparente de plusieurs fournisseurs de services d'application et les vendeurs envers les abonnés mobiles, les entreprises et autres segments verticaux. Il s'agit d'une composante importante de la 5G une architecture qui soutient une variété d'applications innovantes et les services où une latence ultra-faible est requise.

I.6.2 Les caractéristiques de la technologie MEC

- a. **Sur place** : Les plates-formes MEC peuvent fonctionner isolément du reste du réseau, tout en ayant accès aux ressources locales. C'est très important pour les scénarios machine-machine. La propriété de la MEC de s'isoler des autres réseaux la rend également moins vulnérable.
- b. **Proximité** : Déployée à l'endroit le plus proche, l'informatique MEC présente l'avantage d'analyser et de matérialiser des données volumineuses. Elle est également utile pour les appareils gourmands en calcul, tels que la réalité augmentée, l'analyse vidéo, etc.
- c. **Une latence plus faible** : Les services informatiques MEC sont déployés à l'endroit le plus proche des appareils des utilisateurs, isolant ainsi le mouvement des données du réseau central. Ainsi, l'expérience de l'utilisateur est considérée comme de haute qualité avec une latence ultra-faible et une bande passante élevée.

- d. **Sensibilisation à l'emplacement** : Les dispositifs distribués en périphérie utilisent une signalisation de bas niveau pour le partage des informations. La MEC reçoit des informations des dispositifs en périphérie du réseau d'accès local pour découvrir l'emplacement des dispositifs.
- e. **Informations sur le contexte du réseau** : Les applications fournissant des informations de réseau et des services de données de réseau en temps réel peuvent profiter aux entreprises et aux événements en mettant en œuvre la MEC dans leur modèle commercial. Sur la base des informations en temps réel du RAN, ces applications peuvent estimer l'encombrement de la cellule radio et de la largeur de bande du réseau. Cela les aidera à l'avenir à prendre des décisions intelligentes pour une meilleure fourniture de services aux clients.

I.6.3 Architecture MEC

La MEC est une couche qui se situe entre le nuage et les appareils mobiles. Par conséquent, l'infrastructure est dérivée comme une hiérarchie à trois couches : le nuage, la MEC et les appareils mobiles. La figure 1.3 illustre la formation d'un modèle de service à trois niveaux.

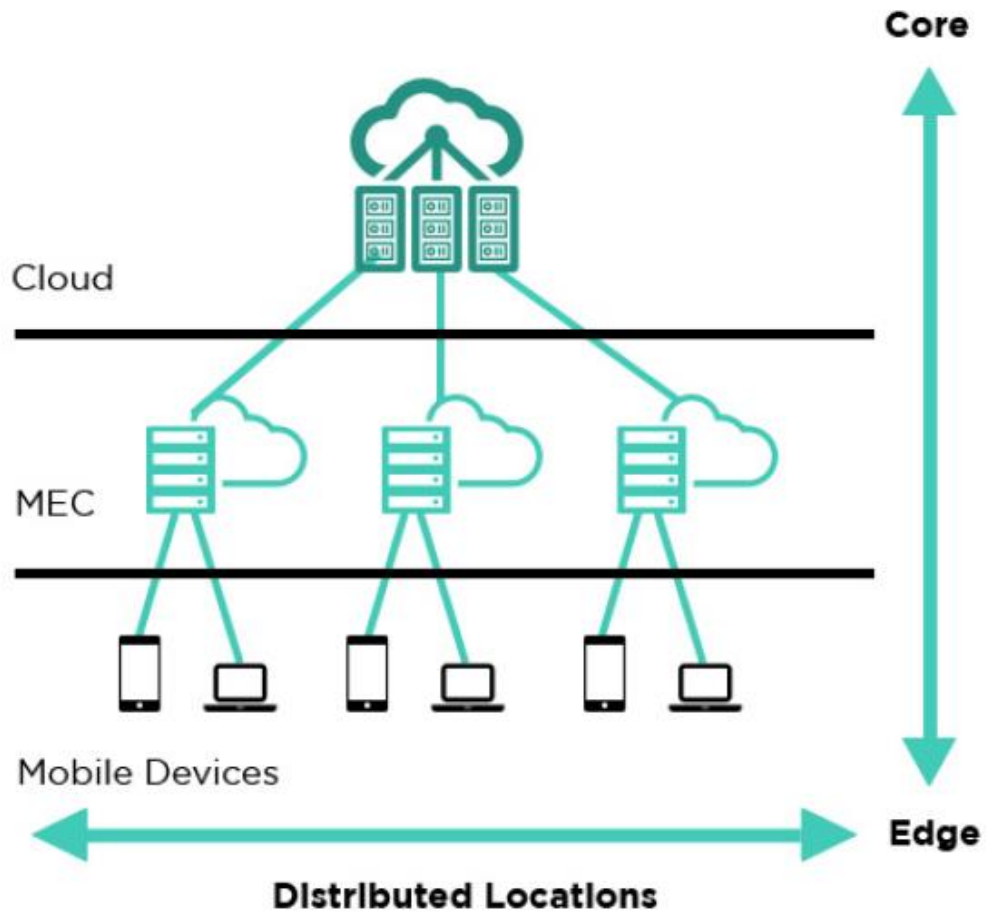


Figure I.4: Modèle de service à 3 niveaux [9].

MEC se conforme principalement à l'informatique dans les nuages pour soutenir et améliorer les performances des appareils finaux.

MEC peut utiliser le réseau cellulaire éléments, tels que la station de base, le point d'accès Wifi ou le femto point d'accès (c'est-à-dire station de base cellulaire de faible puissance).

I.7 Conclusion

Dans ce chapitre, nous avons présenté les différents concepts liés à notre travail. Tout d'abord, nous avons détaillé le streaming vidéo, la qualité de service ensuite la qualité de l'expérience et la relation entre les deux. Nous terminons par introduire l'architecture MEC.

Le but de ce chapitre est de mieux comprendre les notions de base pour atteindre l'objectif de notre travail dans les chapitres suivants qui est l'amélioration de la qualité de l'expérience des services de streaming vidéo adaptatif dans l'architecture MEC.

Chapitre II : État de l'art de vidéo streaming adaptatif

Chapitre II : Etat de l'art de vidéo streaming adaptatif

II.1 Introduction

Aujourd'hui, avec la demande croissante de services multimédias une grande concurrence a été créée entre les fournisseurs de cette catégorie de services, et donc la qualité d'expérience (QoE) devient un facteur déterminant de réussite ou d'échec de ces services et c'est le but de ce travail d'améliorer la qualité d'expérience d'un vidéo streaming adaptatif dans une architecture MEC, prennent en compte les paramètres du terminal final.

Ce chapitre décrit quelques approches qui existent déjà pour améliorer la qualité d'expérience dans les services de vidéo streaming en fonction des paramètres du terminal final.

II.2 Approches récentes pour l'amélioration du streaming adaptatif basées sur l'utilisateur

II.2.1 Approche 1 : Streaming adaptatif dynamique sur HTTP (DASH) [10]

DASH est une norme ISO pour le streaming dynamique adaptatif sur HTTP l'idée principale est la création d'un standard pour les systèmes de streaming adaptatif ayant le potentiel de remplacer toutes les technologies propriétaires existantes comme Microsoft Smooth Streaming, Adobe Dynamic Streaming et Apple HTTP Live Streaming.

a. Mécanisme de travail

La diffusion en continu adaptative définit des segments d'une durée de quelques secondes et un format de liste de lecture qui décrit des codages alternatifs de chaque segment avec différents débits binaires, résolutions, codecs, etc.

Les appareils peuvent télécharger une séquence de segments de types compatibles avec un débit binaire vidéo et une résolution particuliers, et un langage audio à partir d'un ensemble d'alternatives généralement stockées sur des serveurs Web.

DASH utilise XML pour décrire les présentations de médias dans un fichier manifeste(MPD). Il référence les flux de médias stockés dans le format ISO Base Media

File Format. Avec le protocole HTTP et les CDN existants, le DASH permet une meilleure expérience vidéo pour les utilisateurs finaux en s'adaptant automatiquement aux conditions variables du client et du réseau pendant la lecture.

La figure suivante montre clairement l'architecture de l'approche :

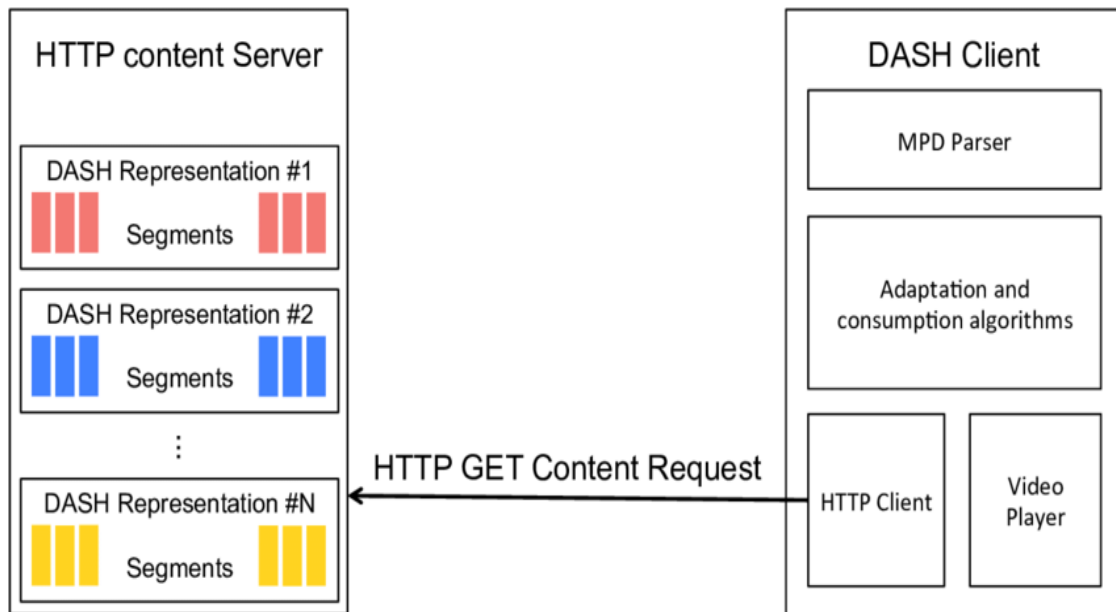


Figure II.1: L'architecture de l'approche [11].

• DASH client

1. Le client télécharge et lit le MPD pour obtenir des informations importantes (l'emplacement du contenu, le codage des segments, la résolution, les largeurs de bande minimales et maximales).
2. Le client sélectionne un encodage de segment approprié et commence à diffuser le contenu en continu par une série de requêtes HTTP. Le serveur crée et encode chaque segment à la demande pour chaque requête.
3. Le client met en mémoire tampon les données au fur et à mesure qu'elles sont téléchargées, tout en gardant une trace des fluctuations de la bande passante de la connexion.
4. Si nécessaire, le client passe automatiquement à un codage de segment différent (parmi ceux répertoriés par le MPD), plus compatible avec le débit binaire actuel.

Cela permet au client de conserver une mémoire tampon suffisante tout au long de la vidéo sans télécharger plus de données que nécessaire.

b. Les algorithmes d'adaptation [5]

La logique de streaming permet une évolutivité et une flexibilité élevées. Par conséquent, les algorithmes d'adaptation de débit ne sont pas standardisés dans DASH, l'objectif étant de choisir un débit garantissant une bonne qualité vidéo et d'éviter les interruptions de lecture vidéo inutiles (Congestion).

La figure suivante montre le Diagramme de flux pour les algorithmes d'adaptation pour DASH.

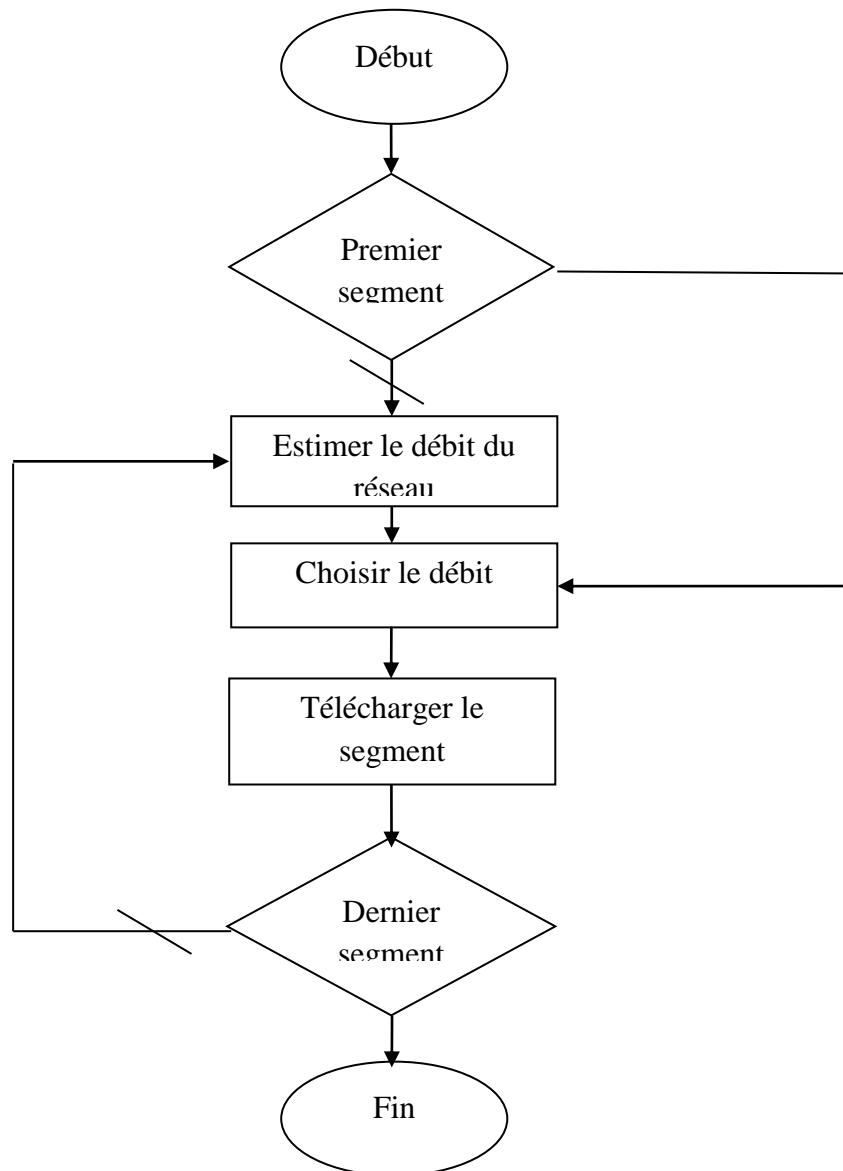


Figure II.2: Diagramme de flux pour les algorithmes d'adaptation pour DASH [5].

c. Les niveaux de DASH [12]

DASH est défini sur deux niveaux :

1. Un cadre générique pour les services streaming adaptatifs dynamiques indépendant de format d'encapsulation de données pour les segments de médias.
2. Une instanciation spécifique de ce Framework avec le format de fichier multimédias de base 3GPP/ISO en spécifiant les formats des segments en partie en se référant au format TS26,244.

d. Etude de l'approche

Après l'étude de cette approche on peut résumer les avantages et les inconvénients comme se suit.

❖ Les avantages

1. S'adapte automatiquement à la bande passante disponible.
 - Choix du débit par le client, en fonction des capacités.
 - Passage transparent d'une qualité à une autre.
2. Déploiement facile sur les CDNs.
3. Nul besoin d'un serveur de streaming.
 - Serveur web .
4. Déplace l'intelligence du réseau vers les clients.

❖ Les inconvénients

1. L'augmentation de latence.
 - les clients doivent mettre en mémoire tampon quelques fragments pour éviter la saturée.
2. Elle utilise que la bande passante et le processeur pour adapter la qualité.
3. La vidéo de sortie doit être mise à l'échelle dans des résolutions adaptées aux appareils clients.

II.2.2 Approche 2 : TCP et l'amélioration des services de vidéo streaming adaptatif [12]

Cette solution consiste à améliorer la qualité d'expérience des utilisateurs de service de vidéo streaming adaptatif en utilisant les paramètres du protocole de transport TCP.

Le principe de cette approche est de calculer un facteur utilisateur globale UF en fonction des paramètres du terminal de l'utilisateur et sera mis à jour en fonction de son

feedback en utilisant l'apprentissage par renforcement. Ce facteur est appliqué par la suite en utilisant le paramétrage TCP et le routage interne afin de recevoir une qualité de vidéo qui satisfait les utilisateurs. Cette approche a été validée par une émulation.

a. Les étapes de cette approche

Le principe de cette solution est de focaliser sur 3 paramètres : **la résolution de l'écran, la taille de l'écran et la batterie disponible.**

❖ Étape 1 :

Cette étape sert à calculer un facteur global de l'utilisateur UF comme suit :

$$UF = FR + FB + FS.$$

Tel que :

FR : le facteur de résolution d'affichage.

'Par exemple, la résolution 720p a comme facteur 4'

FS : le facteur de taille de l'écran.

FB : le facteur de batterie disponible.

La pondération de chaque paramètre va aider à estimer la valeur du facteur global UF .en effet UF est égale à la somme des trois sous-facteurs.

❖ Etape 2 :

Une fois le facteur utilisateur estimé, ils entament l'étape d'estimation de débit. Ils attribuent à chaque facteur d'utilisateur UF une valeur de qualité de vidéo.

Après l'estimation du facteur d'utilisateur et l'attribution d'une qualité vidéo au facteur d'utilisateur, ils attribuent par la suite à cette qualité vidéo à un débit, cette valeur de débit sera utilisée pour limiter le flux vidéo de l'interface Internet.

Ils ont doté leur approche d'un mécanisme d'apprentissage par renforcement, pour une meilleure satisfaction. Ce mécanisme permet de mettre à jour les valeurs de la qualité vidéo attribuées à chaque facteur d'utilisateur UF en fonction du feedback ou de la note donnée au service par l'utilisateur. Cette note est comprise entre 1 et 5, et elle représente le

ressenti global de l'utilisateur ou le MOS (Mean Opinion Score) vis-à-vis du service après chaque visualisation.

Ils vont modéliser le contexte dynamique de la phase d'adaptation des valeurs de la qualité vidéo avec celles du facteur utilisateur et donc L'objectif est de caractériser cette phase d'adaptation par un processus décisionnel de type PDM (Processus de Décision de Markov).

• **Formulation du processus de décision de Markov :**

Un PDM et un ensemble composé de quatre éléments (S ; A ; P ; r) qui intervient dans le processus d'apprentissage par renforcement. Ces composants sont S, l'ensemble des états du système, A un ensemble d'action, P(s) la probabilité de transition de l'état s_t à l'instant t vers l'état s_{t+1} à l'instant t+1, lorsqu'une action a_t est appliquée. La fonction de récompense $r(s)$ indique le retour (positif ou négatif) immédiat lorsque l'action a_t est appliquée à l'état s_t . Au niveau de chaque état du système, un agent calcule le gain obtenu. L'objectif de cet agent est d'apprendre quelle action choisir pour un état s_t afin de maximiser la fonction de récompense cumulative.

Les quatre composants de PDM sont :

- Les états du système.
- Les actions du système.
- La transition des états.
- La fonction de récompense.

L'algorithme suivant représente l'approche généralement :

❖ **Algorithme : adaptation de la bande passante**

- 1- *Début*
- 2- *Tant que [playing.video = 1] faire // l'utilisateur est en train de regarder une vidéo.*
- 3- *{*
- 4- *Wait (300000) ;// Exécuter l'opération chaque 5 min.*
- 5- *Get (TR, BC, SS) ;// obtenir les paramètres du terminal*
- 6- *// Calculer le facteur d'utilisateur*
- 7- *UF = user_factor (TR, BC, SS, Tab_I [], Tab_II [], Tab_III []);*
- 8- *// choisir la qualité vidéo qui va avec le facteur de l'utilisateur.*

```
9- VQ = chose_quality (UF, Tab_V []);
10-// choisir la bande passante qui va avec la qualité vidéo.
11-RB = chose_bdw (VQ, Tab_VI []);
12-// définir RB comme bande passante maximale de l'interface de sortie de
    l'interface sans fil
13- Set (RB, wlan0);
14- }
15- -// après la fin de la vidéo, l'utilisateur donne sont feedback.
16- Get (user_feedback);
17-// Mise à jour des tableaux V, VII par rapport au feedback de l'utilisateur
18-// en utilisant les fonctions d'apprentissage par renforcement.
19- Update (Tab_V [], user_feedback);
20- - Update (Tab_VI [], Tab_VII [], user_feedback);
21- Fin
```

b. Etude de l'approche

Après avoir étudié cette méthode, nous pouvons résumer les avantages et inconvénients comme suit :

❖ Les avantages

1. L'utilisation des paramètres de l'utilisateur final :
 - Résolution d'affichage, taille de l'écran, et batterie disponible.
2. L'utilisation d'apprentissage par renforcement est très importante dans les incertains.

❖ Les inconvénients

1. Le manque de contrôle de congestion :
 - Il est très important pour améliorer la QOE.
2. La solution est décentralisée.
3. Elle ne prend pas d'autre paramètre comme RAM et CPU.

II.2.3 Approche 3 : Amélioration de QOE des services de vidéo streaming adaptatif dans une architecture MEC [5]

Cette approche a été proposée en 2019 vise à améliorer la qualité d'expérience d'utilisateur dans un architecture MEC, elle prend en considération les paramètres d'utilisateur finale tel que résolution de l'écran, la taille de l'écran et la batterie disponible pour calculer le facteur d'utilisateur, donc chaque utilisateur reçoit les qualités de vidéo selon son facteur, la solution a été validé par une émulation.

a. L'architecteur De l'approche

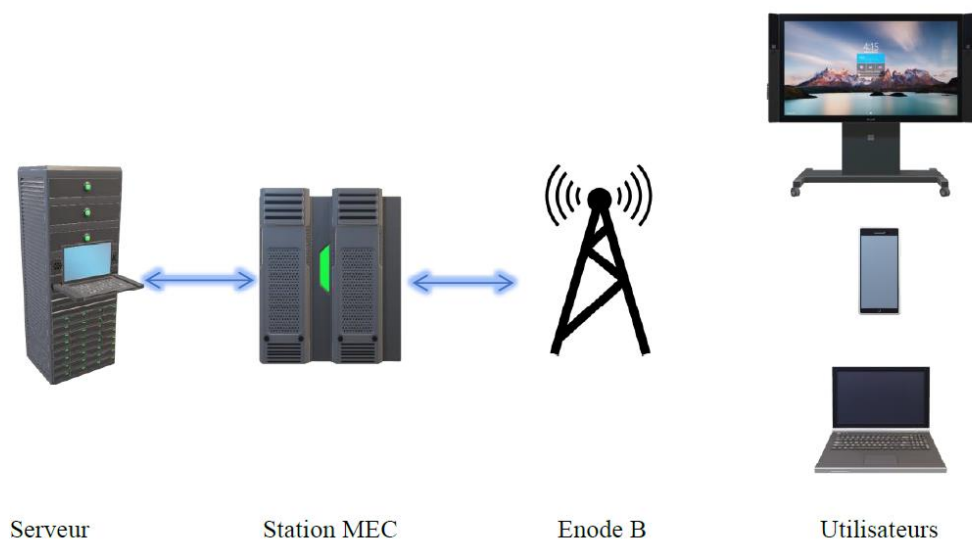


Figure II.3: Architecture de l'approche

- Serveur : c'est un service de streaming vidéo de bout en bout.
- Station MEC : sert à calculer le facteur d'utilisateur.
- Enode B : pour le contrôler en temps réel la congestion.
- Utilisateur : c'est l'utilisateur final qui peut être un pc, un smartphone ou une télévision intelligente.

b. Description de l'approche :

Elle compose de 3 étapes essentielles :

❖ Etape 1 :

Cette étape sert à calculer un facteur d'utilisateur (FU) basé sur certains paramètres (la résolution de l'écran, la taille de l'écran et la batterie). Ce facteur représente une qualité de vidéo. Puis, faire un filtrage des qualités sur un fichier MPD. Le fichier est dynamiquement

envoyé à l'utilisateur durant toute la vidéo. Pour cela, on récupère le fichier de cette réponse HTTP et l'on filtre les qualités selon le facteur-utilisateur.

Le facteur-utilisateur est calculé comme suit : $FU = FR + FE + FB$.

Tel que :

- **FR** : le facteur de résolution d'affichage.
- **FT** : le facteur de taille de l'écran.
- **FB** : le facteur de batterie disponible.

Le tableau suivant présente l'attribution de chaque facteur d'utilisateur à une qualité de vidéo, cette attribution est utilisée pour filtrer les qualités du fichier MPD.

Facteur-utilisateur (FU)	Qualité de vidéo(en pixels)
FU=1.5	360
1.5 <FU <=3	480
3 < FU <= 4.5	720
4.5 <FU	1080

Tableau II-1: Attribution de qualité.

❖ Etape 2 :

Cette étape concerne le contrôle de la congestion elle se fait avant le filtrage pour assurer qu'il n'y aura pas de congestion, si il y en a, les facteurs vont passer à ce contrôle de congestion qui diminue leurs valeurs de ceux qui sont déjà en train de regarder pour dégrader la qualité de la vidéo. Elle se fait dans le cas où il y a une forte possibilité de congestion.

Le calcul du risque de la congestion (RC) : $RC = BPU / BPT$.

Tel que :

- **BPU** : est la somme des bandes passantes des qualités de tous les utilisateurs qui sont en train de regarder la vidéo.
- **BPT** : est la bande passante totale que l'encodeur B peut servir.

Le tableau suivant montre la consommation de la bande passante par l'enode selon le facteur d'utilisateur :

Facteur-utilisateur (FU)	Bande passante Max(en kb/s)
FU=1.5	960
$1.5 < FU \leq 3$	1280
$3 < FU \leq 4.5$	2560
$4.5 < FU$	5200

Tableau II-2: Consommation de bande passante.

La politique de risque de la congestion

Ici il y a deux cas possible :

Si $RC \geq 1$: il y aura un fort risque de congestion ;

Dans ce cas une valeur V qui égale à 0.5 va soustraire de tous les facteurs utilisateurs pour diminuer la PBU et donc le risque de la congestion, V va incrémenter de la même valeur s'il y aura encore un fort risque de congestion.

Sinon

Il y aura un moyen à faible risque de congestion, c'est-à-dire (valider l'attribution des facteurs).

A chaque début du processus, le facteur utilisateur d'origine est réaffecté à tous les utilisateurs. Ceci est fait pour s'assurer que les facteurs utilisateurs sont redistribués après la disparition du risque de congestion afin que les utilisateurs puissent restaurer leur qualité d'origine. Par la suite, les facteurs passent par la politique du RC, pour faire cela, il faut qu'on enregistre les facteurs-utilisateurs réels.

❖ Etape 3 :

Cette étape sert à améliorer l'affectation des facteurs-utilisateurs pour satisfaire les clients. Il est demandé à l'utilisateur d'évaluer son expérience en envoyant son feedback qui est une note d'un à cinq selon sa satisfaction, après un certain temps ajouté par l'administrateur, un processus calcule la moyenne de ces notes (MOS). Puis, l'agent de l'apprentissage par renforcement fait une action 'a' qui sert à ajouter une valeur appelée le

facteur de MOS au facteur d'utilisateur selon l'état 'a' de l'environnement qui est à son tour basé sur la moyenne MOS.

La moyenne de MOS est calculée comme suit : $MOS=S_EVAL/N$.

Tel que :

- **S_EVAL** : est la somme des évaluations.
- **N** : le nombre des utilisateurs qui ont fait l'évaluation.

$$FM=FM+V.$$

Le facteur de MOS est initialisé à 0, et il sera calculé en basant sur les états présentés dans le tableau suivant :

NOTE	V(actions)	ETATS
$4.5 \leq MOS$	-0.5	Risque qu'il y a une congestion
$3 \leq MOS < 4.5$	/	Bon
$MOS < 3$	0.5	Mal

Tableau II-3: MOS selon les états.

La politique de l'apprentissage par renforcement

Si l'état est « mal », donc l'agent rajoute 0,5 à FM.

Si l'état est « bon », donc l'agent ne fait rien.

Si l'état est « Risque qu'il y a une congestion », donc l'agent diminue FM de 0,5.

Le facteur-utilisateur sera calculé dorénavant comme suit : $FU=FU+FM$.

❖ Algorithme : Fonction de Contrôle de congestion

Fonction Contrôle_de_congestion (Table [] : tableau de chaîne de caractère) : chaîne de caractère ;

Var RC, PBT, PBU, V =0,5 : réel ;

N=0 : entier ;

Début

Actualiser_Table (Table []) ; // Fu_m=Fu pour toutes les lignes de la table

PBT ← Obtenir_PBT () ;

$N \leftarrow \text{Calculer_n}(\text{Table } [])$; // Le nombre de tous les utilisateurs qui sont en train de regarder la vidéo et connectés

$\text{PBU} \leftarrow \text{Calculer_PBU}(\text{Table } [])$; // PBU de tous les utilisateurs qui sont en train de regarder la vidéo et connectés

$\text{RC} \leftarrow \text{PBU}/\text{PBT}$;

Tant que ($\text{RC} \geq 1$ et $(\text{PBU}/n) > 1,5$) faire

$\text{MAJ}(\text{Table } [], D)$; // Sert à modifier Fu_m pour que $Fu_m = Fu - V$ de tous les utilisateurs qui sont en train de regarder la vidéo

$N \leftarrow \text{Calculer_N}(\text{Table } [])$;

$\text{PBU} \leftarrow \text{Calculer_PBU}(\text{Table } [])$;

$\text{RC} \leftarrow \text{PBU}/\text{PBT}$;

$V \leftarrow V + 0,5$;

Fin faire ;

$\text{Contrôle_de_congestion} := \text{Table } []$; //Retourner table []

Fin.

c. L'étude de cette approche

Après l'étude de cette approche on peut résumer les avantages et les inconvénients comme ce suit.

❖ Les avantages

1. L'utilisation des paramètres de l'utilisateur final :
 - Résolution d'affichage, taille de l'écran, et batterie disponible.
2. Contrôle de congestion :
 - Réduire les coupures de vidéo qui est très important pour améliorer la qualité d'expérience.
3. C'est une approche centralise :
 - Le concept de la centralisation qui est très nécessaire aussi pour contrôler une congestion ou non.
4. L'utilisation l'apprentissage par renforcement est très importante dans les incertains.

❖ Les inconvénients

1. Elle ne prend pas d'autre paramètre comme RAM et CPU.
2. Plusieurs calculs peuvent augmenter la latence.
3. néglige d'autres facteurs :
 - autres utilisations de la bande passante comme le téléchargement.
 - La bande passante est dédiée seulement à la fourniture de service de vidéo streaming.

II.3 Etude comparative

Dans les trois approches que nous avons vues dans ce chapitre, chaque approche vise à améliorer la qualité de l'expérience des services de streaming vidéo adaptatif basée sur les paramètres du terminal final d'une manière différente.

Si nous adoptons la première approche, le DASH prend en compte l'évaluation à l'aide de métriques de qualité de service (QoS) par opposition à la deuxième et la troisième approche prend en compte l'évaluation à l'aide de la métrique de la qualité de l'expérience (QoE).

- ✓ .la première approche (DASH) est basée sur la bande passante et le processeur, la deuxième et la troisième approche sont basées sur des paramètres plus pertinents tels que la résolution la batterie la taille de l'écran.
- ✓ .Passons à la congestion, qui est un concept très important pour améliorer la qualité de l'expérience qui fait défaut dans la première et la deuxième approche, mais qui est présente dans la troisième approche.

On peut comparer les trois approches d'une manière globale ont basant presque sur toutes les fonctionnalités : paramètres de l'utilisateur, l'architecture et l'évaluation avec des métriques.

Tel que :

- ✓ **Paramètres de l'utilisateur** : Bande passante, CPU, Résolution d'affichage, La batterie, la taille de l'écran.
- ✓ **L'architecture** : soit centralisée ou décentralisée.
- ✓ **L'évaluation** : par une émulation ou bien une simulation.
- ✓ **L'évaluation avec les métriques** : la QoE ou QoS.

Le tableau suivant montre clairement cette comparaison :

Les approches	Paramètre d'utilisateur	L'architecteur	L'évaluation	L'évaluation avec des métriques
Approche1	Bande passante, CPU	Décentralisé	Émulation	QOS
Approche2	Résolution, La batterie, La taille de l'écran.	Décentralisé	Émulation	QOE
Approche 3	Résolution, La batterie, La taille de l'écran.	Centralisé	Émulation	QOE

Tableau II-4: Comparaison des approches.

II.4 Conclusion

Plusieurs approches existant afin d'améliorer la QoE offerte par les applications de communications vidéo interactives sur internet, L'importance d'une telle approche est devenue critique en raison de l'augmentation fulgurante du nombre d'utilisateurs de services multimédias.

Ce chapitre a été concentré sur les approches d'amélioration de la qualité d'expérience dans les services de vidéo streaming.

Dans ce chapitre on a bien étudié et analyser avec détaille les approches qui nous intéressent dans ce travail et tirés leurs avantages et inconvénients. A la fin on a terminé par une étude comparative qui va nous aider par la suite.

**Chapitre III : Etude des
performances de l'approche
AVSCU**

Chapitre III : Etude des performances de l'approche AVSCU

III.1 Introduction

L'idée de concept d'émulation est de reproduire un comportement ou une situation particulière d'un système donné.

Le principal objectif de l'émulation réseau est de proposer un environnement d'expérimentation permettant de tester en temps réel des implémentations réelles de protocoles ou d'applications, de façon à pouvoir évaluer leurs propriétés [13].

Dans ce chapitre nous visons à faire une évaluation de l'approche AVSCU par une émulation, notre objectif est de savoir quelles sont les points faibles et les points forts de cette approche. Pour cela nous avons considéré plusieurs scénarios de tests. Nous commençons par un test sur le risque de congestion et le nombre de coupures, un test sur l'impact de l'approche sur la batterie, un autre sur la consommation de la CPU, un test sur la consommation des données utilisateur (le forfait internet).

III.2 L'environnement de l'émulation

L'implémentation de l'approche AVSCU a été réalisée dans un environnement Linux (Ubuntu).

Ubuntu est un système d'exploitation basé sur la distribution Linux (noyau Linux), il est utilisé par des millions de PC à travers le monde avec une interface « simple, intuitive, et sécurisée » Elle est la distribution la plus consultée sur Internet d'après le site Alexa[14].

Le service de streaming choisi est le service DASH car il est open source avec tous ces outils comme l'encodage de la source multimédia, il échange les données avec les requêtes et les réponses http.

III.2.1 La topologie d'émulation

- **PC 1** : c'est un PC portable de la marque HP ProBook, qui agit comme la station MEC et l'encodeur B (point d'accès) de la topologie de l'approche.

- **PC 2** : c'est un PC portable de la marque DELL, agit comme le serveur de la topologie de l'approche.

Le tableau suivant représente les caractéristiques des deux PCs :

	Processeur	Capacité de la RAM	Capacité du disque dur	Type de système
PC1	Intel Core i5-7300U 2.60 GHz	8 GO.	237 GO.	64 bits.
PC2	Intel Celeron 2957 U 1.40 GHz	2 GO	149GO	64 bits

Tableau III-1: Caractéristiques de chaque PCs.

Les terminaux : Nous avons considéré trois types de terminaux : une Smart TV, un ordinateur portable et un Smartphone. Les caractéristiques de chaque terminal sont mentionnées sur le tableau suivant :

Terminal	Taille d'écran	Résolution maximale	Niveau de la Batterie
Smart TV	40 pouces	2160 p	/
Ordinateur Portable	15 pouces	1440 p	/
Smartphone	6 pouces	1280 p	100%

Tableau III-2: Paramètres des terminaux utilisés dans l'émulation.

III.3 Les scénarios du test

Afin d'évaluer l'approche AVSCU, nous avons choisi d'appliquer les scénarios suivants :

III.3.1 Scénario 1 : contrôle de la congestion

Nous avons proposé ce scénario afin de savoir l'efficacité de la fonction de contrôle de la congestion. D'abord on lance la vidéo dans les deux terminaux le Pc et le Smartphone, ensuite on calcule le risque de congestion dans plusieurs reprises en modifiant le seuil de la

bande passante de l'interface du point d'accès pour créer des congestions. Les utilisateurs vent regardé la vidéo au même temps avec leurs terminaux (le pc, Smartphone).

On utilisant la formule suivante pour calculer le risque de la congestion(RC) :

$$RC = BPU / BPT .$$

Tel que :

❖ **BPU** → est la somme des bandes passantes des qualités de tous les utilisateurs qui sont en train de regarder la vidéo.

❖ **BPT** → est la bande passante totale que l'énode B peut servir.

Après l'exécution de ce scénario on a obtenu les résultats suivants:

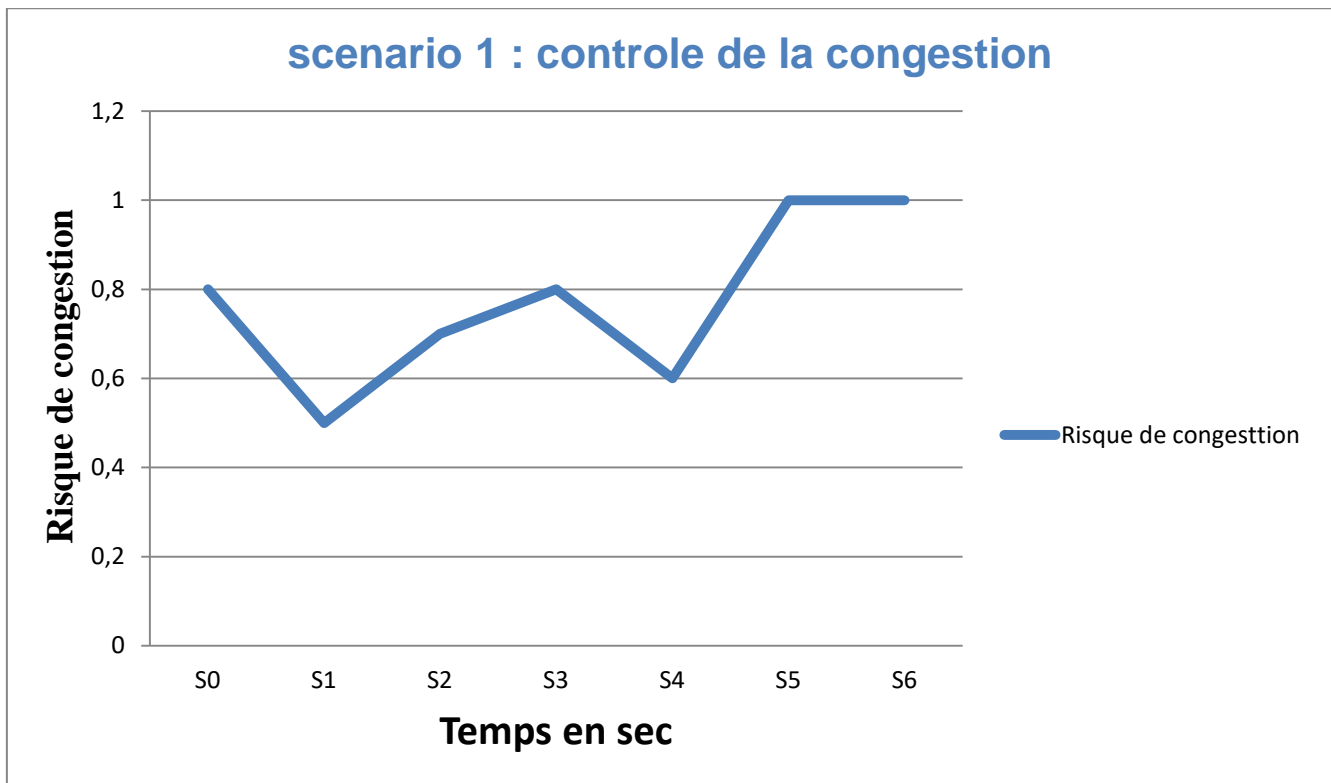


Figure III.1: Résultat de scénario 1.

S0: seuil =12000 kb/s (RC = (5200+5200)/12000=0.8 pas d'interférence de la fonction de contrôle de congestion).

S1 : seuil =10000 kb/s (RC=5200+5200)/10000=1.04 après l'interférence de la fonction de contrôle de congestion RC =0.5).

S2 : seuil =5000 kb/s ($RC = (2650+2650)/5000=1.06$ après l'interférence de la fonction de contrôle de congestion $RC =0.7$).

S3 : seuil =3000 kb/s ($RC = (2650+1280)/3000= 1.3$ après l'interférence de la fonction de contrôle de congestion $RC = 0.8$)

S4 : seuil =2500 kb/s ($RC = (1280+1280)/ 2500=1.02$) après l'interférence de la fonction de contrôle de congestion $RC = 0.6$)

S5: seuil =1200 kb/s ($RC = (960+960)/1200=1.6$ pas d'interférence de la fonction de contrôle de congestion).

S6: seuil =1000 kb/s ($RC = (960+960)/1000=1.9$ pas d'interférence de la fonction de contrôle de congestion)

Résultats et discussion :

D'après le graphe il est clair que l'approche AVSCU en général contrôle la congestion, mais il reste deux cas à S5 et S6 où le risque de congestion arrivera jusqu'à 1.6 et 1.9 respectivement car la fonction du contrôle de congestion ne peut pas éliminer ce fort risque de congestion parce qu'elle a affecté 360p comme qualité aux terminaux, cette qualité ne peut pas être filtrée, de plus la formule qui calcule le risque de la congestion ne présente pas une probabilité car elle dépasse le 1.

Donc l'approche AVSCU ne peut pas éliminer la congestion d'une façon définitive.

III.3.2 Scenario 2 : nombre des coupures

Dans ce test on calcule le nombre des coupures lorsque deux utilisateurs regardent la vidéo au même temps (un pc et un Smartphones) sans et avec l'approche AVSCU, on modifie le seuil de la bande passante de l'interface du point d'accès, pour la raison de créer des congestions. Par la suite Nous avons calculé le totale des coupures dans l'un des deux terminaux (Smartphone).

Les résultats de ce test sont présentés par le graphe suivant:

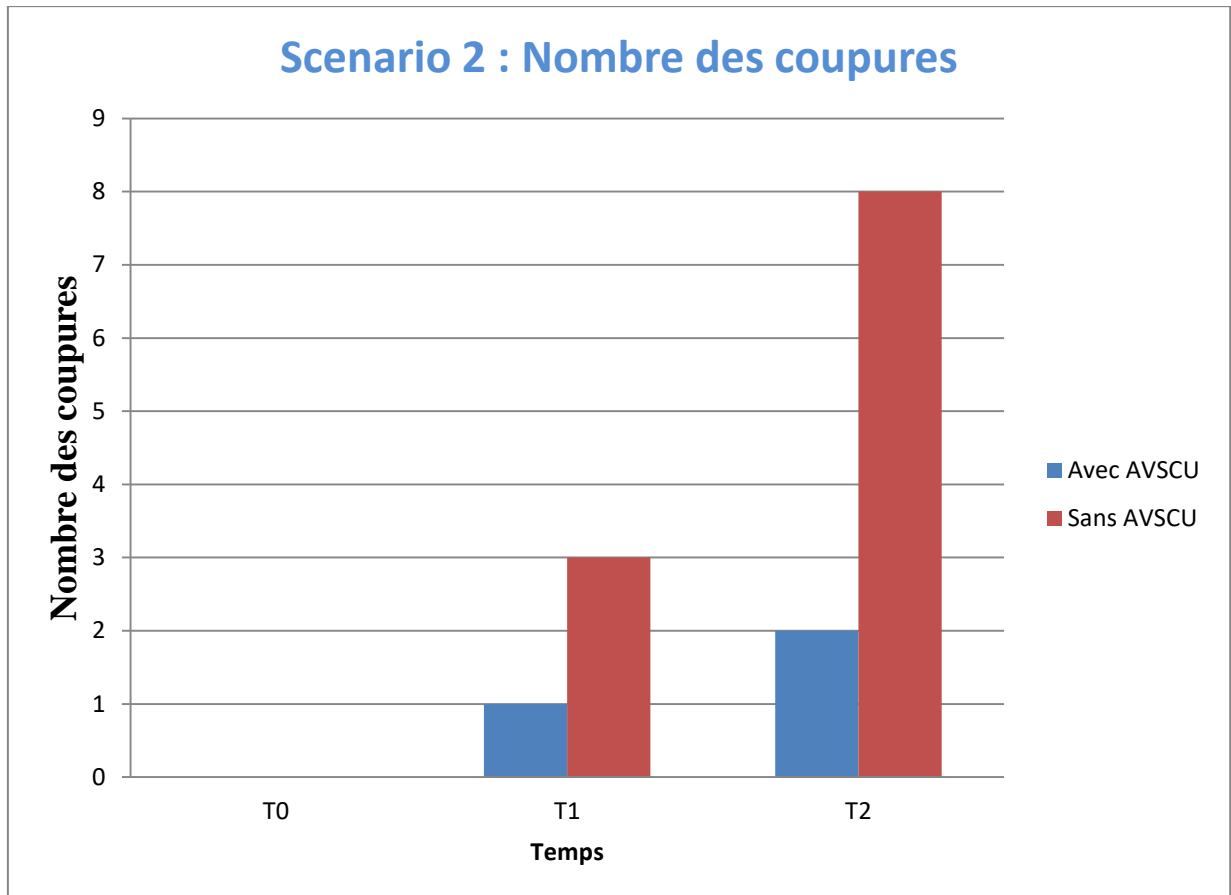


Figure III.2: Résultat de scénario 2.

T0 : le seuil = 15000kb/s.

- Nombre des coupures avec AVSCU = 0
- Nombre des coupures sans AVSCU = 0

T1 : le seuil = 6500kb/s.

- Nombre des coupures avec AVSCU = 1
- Nombre des coupures sans AVSCU = 3

T2 : le seuil = 2500kb/s.

- Nombre des coupures avec AVSCU = 2
- Nombre des coupures sans AVSCU = 8

Résultats et discussion:

A partir du graphe, nous avons remarqué qu'à T0 dans les deux cas "sans et avec l'approche" le nombre des coupures égales à zéro, les qualités reçues par les deux

terminaux sont : 1080p, 720p, 480p, 360p parce que la bande passante a été suffisante pour servir ces qualités, aucune congestion donc pas des coupures.

À T1 avec l'exécution de l'approche AVSCU on a une seule coupure parce que les qualités reçues par les 2 terminaux sont 720p, 480p, 360p, ça grâce à la fonction de contrôle de la congestion qui a diminué les qualités. Par contre dans le cas où on n'a pas utilisé l'approche AVSCU, les qualités reçues sont 1080p, 720p, 480p, 360p par les deux terminaux ce qui provoque plus des coupures.

À T2 sans l'exécution de l'approche AVSCU le nombre des coupures grand par rapport au nombre des coupures si on a utilisé l'approche AVSCU. Malgré le seuil est très petit l'approche AVSCU à pouvoir éviter le maximum des coupures puisque la fonction de la congestion diminuée les qualités (480 p, 360 p pour les deux terminaux), en outre les qualités reçues sans l'approche sont : 1080 p, 720 p, 480 p, 360p ce qui entraînait un nombre énorme des coupures.

On prend comme conclusion que grâce aux algorithmes de l'approche AVSCU on peut regarder une vidéo avec moindre de coupure.

III.3.3 Scenario 3 : consommation de la batterie

Dans ce scénario nous sommes intéressés à la consommation de la batterie lorsqu'un utilisateur regarde la vidéo avec son Smartphone. Ce scénario est divisé en deux parties, la première partie l'utilisateur va regarder la vidéo avec l'exécution de l'algorithme d'adaptation de l'approche AVSCU, dans la deuxième partie l'utilisateur va regarder la vidéo sans l'utilisation l'approche.

On a mesuré le pourcentage de la batterie pendant qu'il regarder la vidéo et on a obtenu les résultats suivants :

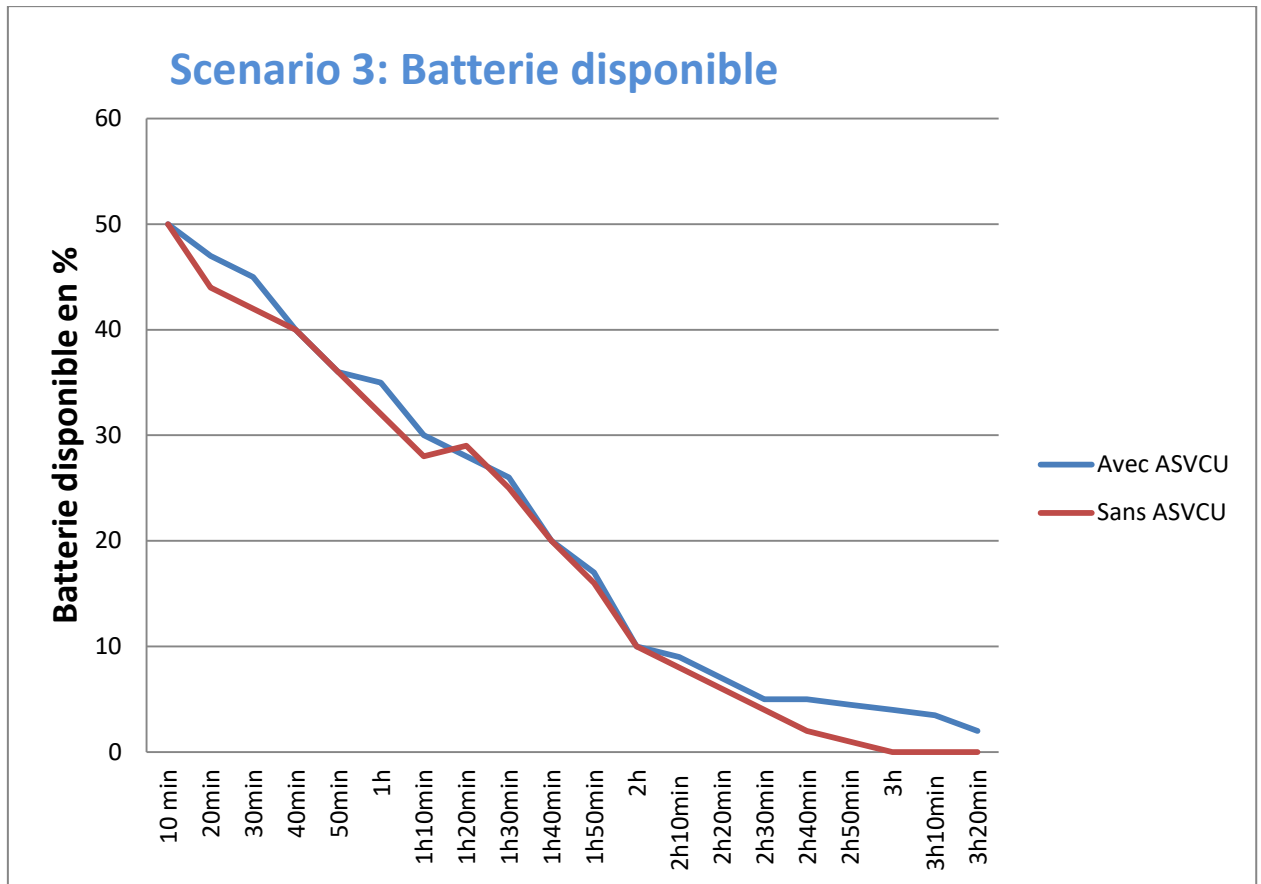


Figure III.3: Résultat de scénario 3.

Résultats et discussion :

Concernant la batterie le graphe montre que lorsque nous regardons la vidéo avec l'approche (AVSCU) la consommation de la batterie est la même il n'a y pas une grande différence si on regardant la vidéo sans l'approche (AVSCU) étant donné que les qualités perçue chez l'utilisateur sont : 1080p, 720p, 480p, et 360p et ça consomment plus de batterie, mais lorsque la batterie arrivera à 5% l'approche AVSCU diminuait les qualités ce qui économiser la consommation de cette dernière, les qualités reçues par l'utilisateur sont : 720p, 480p, 360p.

On peut dire que ce n'est pas vraiment efficace car la charge de la batterie est très basse le Smartphone va s'éteint à n'importe quel moment, l'approche AVSCU aurait pu diminuer les qualités quand la batterie arrivera à 10% ou plus.

III.3.4 Scénario 4 : consommation du CPU

Dans ce scénario nous avons testé la consommation du CPU tandis qu'un utilisateur regarde la vidéo sur son Pc, ce scénario va se faire dans deux cas, le premier cas où on utilise l'approche AVSCU pour regarder la vidéo, le deuxième cas sans l'utilisation de l'approche AVSCU, en utilisant la commande « Htop » pour mesure l'utilisation du CPU comme montre la figure suivante :

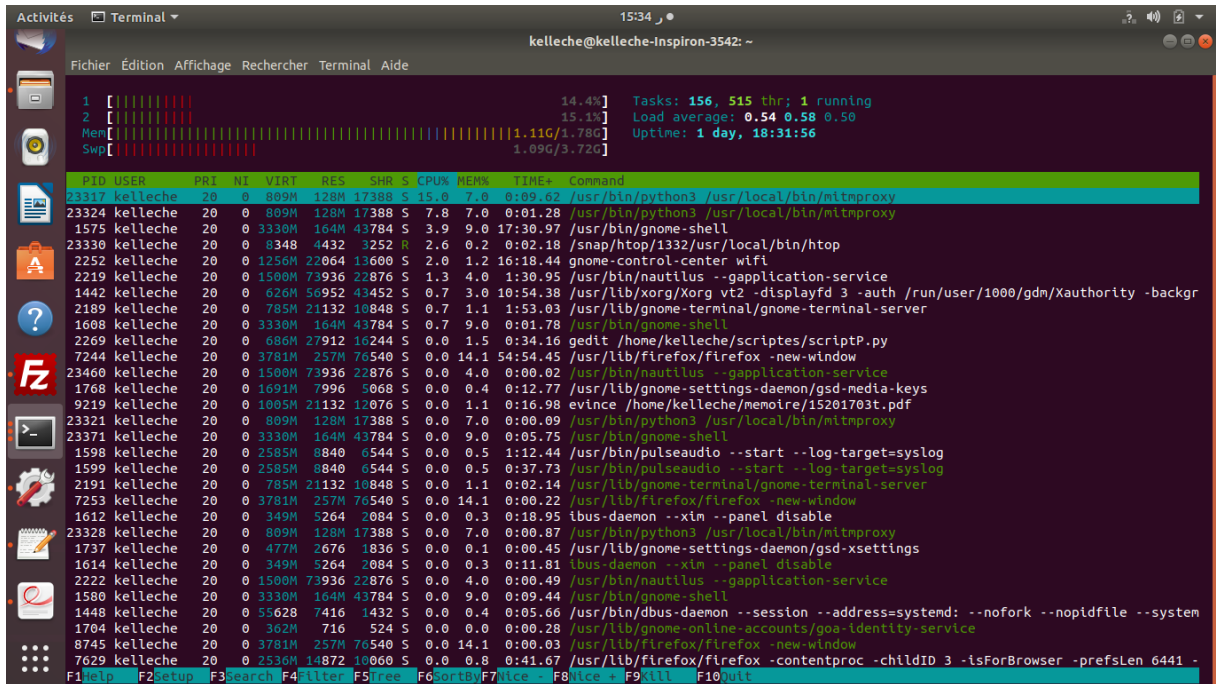


Figure III.4: L'exécution de la commande « htop ».

Après l'exécution de ce scénario on a obtenu des résultats représenté par le graphe suivant :

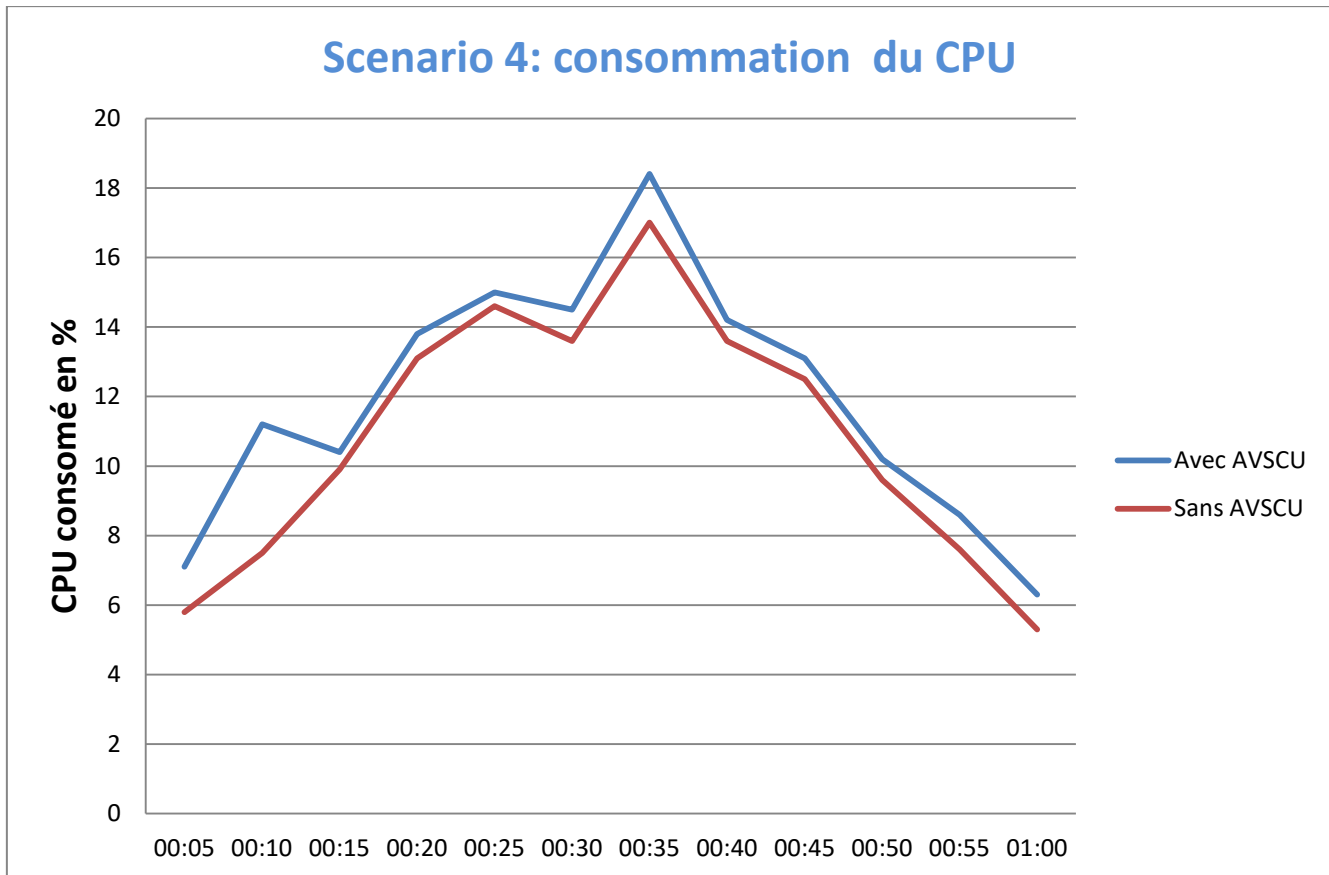


Figure III.5: Résultat de scénario 4.

Résultats et discussion :

Selon le graphe qu'on a obtenu la consommation du CPU avec l'utilisation de l'approche AVSCU est plus élevée que la consommation du CPU sans l'utilisation de l'approche AVSCU, cela est dû au fait que l'approche AVSCU utilise le CPU pour les calculs de facteur d'utilisateur, l'affectation des qualités, le filtrageetc. Le pique qu'on a obtenue à 00 :10 revient au premier calcul du facteur d'utilisateur.

On peut conclure que l'approche AVSCU ne prend pas le CPU en considération avec les autres paramètres d'utilisateur, il n'a y pas une adaptation pour la consommation du CPU.

III.3.5 Scénario 5 : consommation du forfait internet

Afin de tester la consommation des données internet, nous avons proposé ce scénario, l'utilisateur va regarder la vidéo avec son Smartphone sans et avec l'approche AVSCU, on modifie le seuil de la bande passante de l'interface du point d'accès, pour avoir différentes possibilités de congestion.

Après l'exécution de ce scénario on a obtenue des résultats suivants :

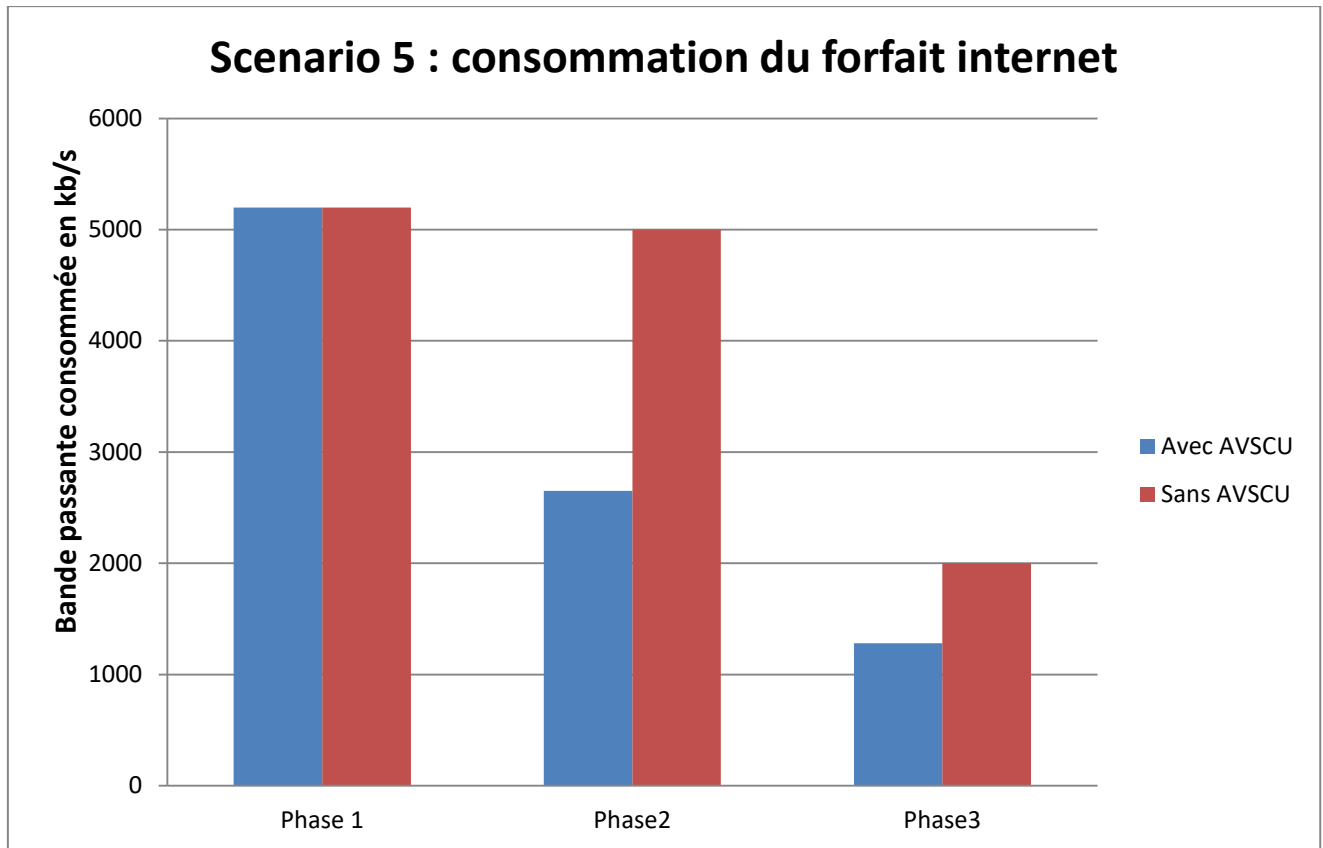


Figure III.6: Résultat de scénario 5.

Phase 1 : le seuil = 6000 kb/s.

- ✓ La bande passante consommée avec l'approche = 5200kb/s.
- ✓ La bande passante consommée sans l'approche = 5200kb/s.

Phase 2: le seuil=5000 kb/s.

- ✓ La bande passante consommée avec l'approche = 2650kb/s.
- ✓ La bande passante consommée sans l'approche = 5000kb/s.

Phase 3 : le seuil = 2000 kb/s.

- ✓ La bande passante consommée avec l'approche = 1280kb/s.
- ✓ La bande passante consommée sans l'approche = 2000kb/s.

Résultats et discussion :

A partir de ce graphe, nous avons observé que lorsque la bande passante est suffisante comme dans la première phase, la consommation de cette dernière est totale car les qualités reçues par l'utilisateur sont: 1080p 720p 480p 360p qui consomment respectivement : 5200

kb/s 2650 kb/s 1280 kb/s 960 kb/s donc la plus haute qualité que l'utilisateur peut choisir est 1080p qui consomme environ 5200 kbps sans et avec l'approche AVSCU.

Dans la phase 2 on remarque que la consommation de la bande passante n'est pas la même, si on regardant la vidéo sans l'exécution du script les qualités reçues par l'utilisateur sont : 1080p, 720p, 480p, 360p la plus grande qualité qui peut être choisi par l'utilisateur est 1080p qui consomme environ 5200 kb/s donc la consommation de la bande passante est totale pour servir ce besoin par contre si on exécutant le script de l'approche AVSCU les qualités reçue par l'utilisateur sont 720p, 480p, 360p, donc la plus grande qualité qui peut être choisit par l'utilisateur est 720p qui consomme environ 2650 kb/s ce qui va économiser la bande passante .

La même chose pour la phase 3, sans l'utilisation du script l'utilisateur reçoit toutes les qualités, la plus grande qualité est 1080p donc il va consommer toute la bande passante pour servir ce besoin en revanche, en exécutant le script de l'approche AVSCU, l'utilisateur ne reçoit pas toutes les qualités, la plus grande qualité affectée et 480p qui consomme environ 1280 kb/s c'est bien pour économiser son forfait internet.

À partir de ces résultats, on s'en sort avec un point très important. L'approche AVSCU économisé la consommation de la bande passante grâce au contrôle de la congestion qui diminue les qualités selon la bande passante disponible.

III.4 Récapitulatif des scénarios

Chacun de ces tests nous montre soit un point fort, soit un point faible pour l'approche AVSCU, à commencer par le point fort, on peut dire que cette approche a un effet sur la consommation de bande passante du fait de la fonction de contrôle de congestion qui réduit les qualités qui économisent la consommation des forfait Internet, comme l'a montré le cinquième test. Le deuxième test nous montre que grâce à cette approche, nous pouvons regarder une vidéo avec moins de coupures, ce qui est très demandé par les clients, c'est donc un point fort. En ce qui concerne la congestion le premier test montre que l'approche AVSCU peut contrôler la congestion et ça un point très important pour éviter les coupures mais la formule qui calculer le risque de la congestion ne présenter pas une probabilité. Ces tests nous ont également permis d'identifier les points faibles. Le troisième test de consommation de la batterie montre que lorsque la batterie atteint 5% l'approche AVSCU diminue les qualités qui réduisent la consommation de la batterie mais n'est pas efficace aurait pu mieux économiser la consommation de la batterie. Le quatrième test concernait la

consommation CPU ce test montre que l'approche AVSCU ne prend pas en compte le CPU.

III.5 Conclusion

Au cours de ce chapitre, nous venons d'étudier en détails les performances de l'approche AVSCU, nous avons analysé chaque test pour l'extraction des forces et faiblesses afin de nous guider dans l'amélioration de cette approche qui sera présentée dans le prochain chapitre.

**Chapitre IV : Conception d'un
mécanisme d'amélioration de la
QoE dans les services de vidéo
streaming adaptatif**

Chapitre IV : Conception d'un mécanisme d'amélioration de la QoE dans les services de vidéo streaming adaptatif

IV.1 Introduction

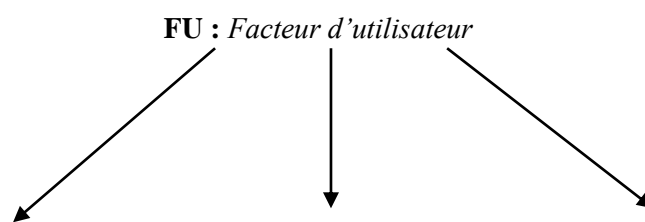
Dans ce chapitre, nous présenterons des améliorations pour l'approche AVSCU qui se base sur les paramètres du terminal dans l'architecture MEC.

Ce chapitre comprend d'abord le mécanisme du fonctionnement, puis nous présenterons les améliorations d'adaptabilité (la consommation de la batterie et la formule de la congestion). Enfin, nous terminons par l'algorithme de la solution et son organigramme.

IV.2 Mécanisme du fonctionnement

Lorsqu'un utilisateur souhaite regarder une vidéo avec l'architecture MEC, il doit d'abord se connecter au serveur via le point d'accès (l'node B), avant que l'utilisateur reçoit les qualités la station MEC doit être en mesure de calculer le facteur de chaque utilisateur en fonction des paramètres pertinents de ce qui demande la vidéo, Par conséquent, chaque utilisateur obtiendra la qualité vidéo en fonction de ses propres facteurs. La station MEC peut également contrôler la congestion au niveau de l'node en temps réel.

Le facteur utilisateur est calculé selon la formule suivante :



$$\mathbf{BD} : \text{batterie disponible} + \mathbf{RT} : \text{Résolution de l'affichage} + \mathbf{TE} : \text{Taille de l'écran} \quad (1)$$

IV.3 Les étapes de la solution proposée

❖ Etape 1 : Adaptation de la batterie

Cette étape permet de calculer le facteur utilisateur en fonction du poids de certains paramètres (tels que la résolution de l'écran, la taille de l'écran et la batterie) envoyés par l'utilisateur en tant que requête HTTP. Ce facteur représente la qualité vidéo. Ensuite, filtrez la qualité du fichier MPD. Le fichier sera envoyé dynamiquement à l'utilisateur tout au long de la vidéo. À cette fin, nous avons obtenu le fichier de réponse HTTP et filtré la qualité en fonction des facteurs utilisateur.

Dans cette partie, nous proposerons des améliorations de la consommation de la batterie, comme nous l'avons déjà vu dans le chapitre précédent. Nous avons constaté que la consommation de la batterie en utilisant l'approche AVSCU est approximativement la même que la consommation de la batterie sans l'approche AVSCU (voir scénario 3 : consommation de la batterie). Par conséquent, après avoir obtenu ce résultat, nous essaierons de prêter plus d'attention à la batterie, Cela se fait en donnant plus d'importance à la batterie plutôt qu'à l'affichage et à la résolution. Ainsi, nous proposons de modifier les pondérations de ces paramètres comme suit :

FB : le facteur de batterie disponible, calculé sur la base du *Tableau IV-1*.

FR : le facteur de résolution d'affichage, calculé sur la base du *Tableau IV-2*.

FT : le facteur de taille de l'écran, calculé sur la base du *Tableau IV-3*.

BD : Batterie disponible en %	FB
BD <= 25	1
25 < BD <= 50	2
50 < BD <= 75	3
75 < BD	4

Tableau IV-1: Valeur de FB.

RT : Résolution de l'affichage en pixels	FR
RT <= 360	0.25
360 < RT <= 480	0.5
480 < RT <= 720	0.75
720 < RT	1

Tableau IV-2: Valeur de FR.

TE: Taille de l'écran en pouces	FT
TE <= 7	0.25
7 < TE <= 13	0.5
13 < TE <= 20	0.75
20 < TE	1

Tableau IV-3: Valeur de FT.

Les attributions des qualités et de la bande passante selon le facteur d'utilisateur sont représentées par le tableau suivant :

Facteur-utilisateur (FU)	Bande passante Max (en kb/s)	Qualité de vidéo (en pixels)
FU = 1,5	960	360
1,5 < FU <= 3	1280	480
3 < FU <= 4,5	2650	720
4,5 < FU	5200	1080

Tableau IV-4: L'attribution de la qualité [5].

❖ Etape 2 : la formule de la congestion

La bande passante de la station MEC peut être utilisée par l'encodeur B (utilisé aujourd'hui comme point d'accès du réseau LTE) à travers sa bande passante pour connecter les utilisateurs, Cela nous donne la possibilité de contrôler la congestion du trafic de la vidéo en streaming, dans le cas où il y a une congestion, les facteurs vont passer à ce contrôle de congestion qui diminue leurs valeurs de ceux qui sont déjà en train de regarder pour dégrader la qualité de vidéo, cette étape est faite avant le filtrage des qualités pour assurer

son efficacité, réduisant ainsi les coupures vidéo, ce qui est très important pour améliorer la qualité de l'expérience .

Comme nous l'avons déjà vu dans le chapitre précédant la formule qui calcule le risque de la congestion ne présente pas une probabilité car elle dépasse le 1 donc dans cette partie et afin de calculer le risque de la congestion, nous proposons une formule de probabilité de congestion basée sur le flux sortant de l'interface de l'enode B, qui inclut non seulement le flux vidéo mais également d'autres flux (tels que les paquets de données IP), et la capacité totale de l'interface.

La formule de probabilité de congestion est exprimée comme suit :

$$\text{Proba} = \text{flux_inte} / \text{capacité_inte} \quad (2)$$

Tel que :

- **Flux_inte** : le flux sortant de l'interface de l'enode B qui couvert ne pas seulement le flux de streaming mais tous les flux sortant de cet interface.
- **Capacité_inte** : la capacité totale qui peut servir l'interface de l'enode B.

Les tableaux suivants montrent l'attribution des qualités selon la probabilité et le facteur d'utilisateur :

Proba	Bande passante max(en kb/s)
0<proba=<0.7	5200
0.7<proba=<0.8	2560
0.8<proba=<0.9	1280
0.9<proba=<1	960

Tableau IV-5 : L'attribution de la qualité.

FU	Bande passante max (en kb/s)
FU = 1,5	960
1,5 < FU <= 3	1280
3 < FU <= 4,5	2560
4,5 < FU	5200

Tableau IV-6: La consommation de la bande passante [5].

La politique de la Probabilité est :

- Si $\text{Proba} \geq 0.7$ l'algorithme de la fonction de congestion va soustraire une valeur appelée V tel que $V=0.5$ de tous les facteurs utilisateurs pour la raison de diminuer la bande passante d'utilisateur et donc la probabilité de congestion, V va

incrémenter de la même valeur s'il y aura encore une forte probabilité de congestion.

- Sinon l'algorithme va valider l'attribution des facteurs.

À chaque début du processus, le facteur utilisateur original est réaffecté à tous les utilisateurs. Ceci est fait pour garantir que les facteurs utilisateurs sont redistribués lorsque la possibilité de congestion disparaît afin que les utilisateurs puissent restaurer leur qualité d'origine. Par la suite, les facteurs doivent passer par une stratégie de probabilité, pour cela, il faut enregistrer les facteurs réels de l'utilisateur.

Algorithme de Contrôle de congestion

Fonction Contrôle_de_congestion (Table [] : tableau de chaîne de caractère) : chaîne de caractère ;

Var Proba, PBT, PBU, V =0,5 : réel ;

N=0 : entier ;

Début

Actualiser_Table (Table []); // Fu_m=Fu pour toutes les lignes de la table

PBT ← Obtenir_PBT ();

N ← Calculer_n (Table []); // Le nombre de tous les utilisateurs qui sont en train de regarder la vidéo et connectés

PBU ← Calculer_PBU (Table []); // PBU de tous les utilisateurs qui sont en train de regarder la vidéo et connectés

used_rate ← Obtenir_used_rate (); //le flux sortant de l'interface de l'inode B

Proba ← used_rate/PBT;

Tant que (Proba >= 0,7 et (PBU/n) > 1,5) faire

MAJ (Table [], D); // Sert à modifier Fu_m pour que Fu_m= Fu - V de tous les utilisateurs qui sont en train de regarder la vidéo

N ← Calculer_N (Table []);

PBU ← Calculer_PBU (Table []);

Proba ← used_rate/PBT;

V ← V+ 0,5 ;

Fin faire ;

Contrôle_de_congestion := Table [] ; //Retourner table []

Fin.

IV.4 Conclusion

Ce chapitre a été concentré sur la conception de notre solution visant à améliorer l'approche AVSCU (consommation de la batterie, formule de congestion) et donc à améliorer la qualité de l'expérience du vidéo streaming adaptatif dans une architecture MEC.

Le chapitre prochain concerne la mise en œuvre et l'évaluation de notre solution, testons notre solution avec des scénarios pour montrer l'efficacité.

Chapitre V : Implémentation et étude expérimentale

Chapitre V : Implémentation et étude expérimentale

V.1 Introduction

L'objectif principal de ce chapitre est l'implémentation d'une solution vis à améliorer la qualité de l'expérience des services de vidéos streaming dans une architecture MEC en se basant sur les paramètres du terminal final, afin de réduire la consommation énergétique « consommation de la batterie » et d'améliorer le mécanisme du contrôle de la congestion.

Ce chapitre implémente les algorithmes proposés dans le chapitre précédent. Nous commençons par décrire les différents outils d'implémentation de notre proposition. Suivie d'une étude expérimentale et discussion pour les différents scénarios. Nous terminons par une conclusion.

V.2 Outils d'implémentation



Figure V.1: Topologie du streaming.

Le Pc qui agit comme le serveur (voir tableau III.1) contient :

Enp7s0: Il s'agit de l'interface Ethernet qui connecte le serveur à la station MEC.

Serveur DASH [15]: Contient un fichier MPD représente sous forme de vidéo avec 4 qualités différentes et un Fichier audio afin de proposer aux utilisateurs la vidéo à la demande.

(Qualité, Bande passante (kb/s)) : (1080, 4200), (720, 2200), (480, 1060), (360, 600).

(Bande passante d'audio (kb/s)) : (128).

Ces fichiers ont été créés par:

FFMPEG [16]: Afin d'encoder un fichier source vidéo en différentes qualités vidéo et un seul fichier audio.

```
sudo add-apt-repository ppa:mc3man/trusty-media
sudo apt-get update
sudo apt-get install ffmpeg gstreamer0.10-ffmpeg
which ffmpeg /usr/bin/ffmpeg
```

MP4Box-GPAC [17]: Y compris la division du fichier encodé en petits morceaux (segments) et la création d'un fichier manifeste (MPD) pour lier les fichiers segmentés.

```
sudo apt-get install gpac
which MP4Box /usr/bin/MP4Box
```

Le site web : est une page simple qui affiche un lecteur DASH fourni par DASH.js [18].

Le lecteur DASH est un lecteur de vidéo MPEG-DACH, il est open source et écrit en JavaScript, son but est de fournir un lecteur puissant. Il est hébergé par le serveur apache [19].

Les utilisateurs du site doivent pouvoir visionner la vidéo, modifier sa qualité et d'évaluer L'expérience.

Les qualites : ▾

360p



Évaluez votre expérience de 1 à 5 :

La note : 1

Figure V.2: La page du site web.

Le PC qui sert de la station MEC et l'enode B (voir Tableau III.1) comprend:

Enp0s31f6 : c'est l'interface Ethernet relie la station MEC avec le serveur.

Wlp2s0 : C'est l'interface sans fil de l' EnodeB accessible par le client.

WonderShaper [20] : Cet outil est disponible pour les ordinateurs basés sur Linux. Il permet d'ajuster la bande passante disponible sur l'interface à une valeur bien définie.

```
Sudo wondershaper -a <network interface> -d <downspeed> -u <upspeed>
```

Les valeurs «downspeed» et «upspeed» indiquent la bande passante disponible pour télécharger et envoyer en kbit.

MITMProxy [21] : est un proxy intermédiaire interactif avec une interface de console. Il peut intercepter les requêtes et les réponses HTTP et HTTPS, et modifiez-les instantanément.

```
sudo apt install python3-pip  
sudo pip3 install -U pip  
sudo pip3 install mitmproxy
```

Cron [22]: C'est un gestionnaire de tâches effectuées à un moment précis. Chaque utilisateur dispose d'un fichier crontab qui lui permet de signaler l'opération à effectuer souvent.

Les Scripts : Nous choisissons le langage python [23] comme langage de programmation pour mettre en œuvre notre solution proposé.

La solution se compose de 2 scripts. Le premier script « scriptP.py » exécuté par MITMProxy, c'est l'algorithme principal de la solution.

Il effectue les opérations suivantes :

- Apes avoir les paramètres utilisateurs à partir la réponse HTTP ce script Calcule le facteur d'utilisateur.
- L'exécution de la fonction de contrôle de congestion plus le filtrage des qualités vidéo.
- Enregistre l'évaluation de l'utilisateur lors de l'envoi de la requête HTTP de ce dernier.

La fonction MOS a été complétée par le second Script « mos.py » qui fait que le calcul du facteur de MOS.

V.3 Les Scénarios du test

Afin d'évaluer notre solution proposé et de prouver les améliorations qu'elles apportent, Nous avons choisi d'appliquer l'ensemble des tests suivants :

V.3.1 Scénario 1

Ce scénario a pour objectif de voir la consommation de la batterie lorsqu'un utilisateur regarde une vidéo sur son Smartphone, Avec et sans notre solution. Nous avons mesuré le pourcentage de la batterie, les résultats sont représentés par le graphe suivant :

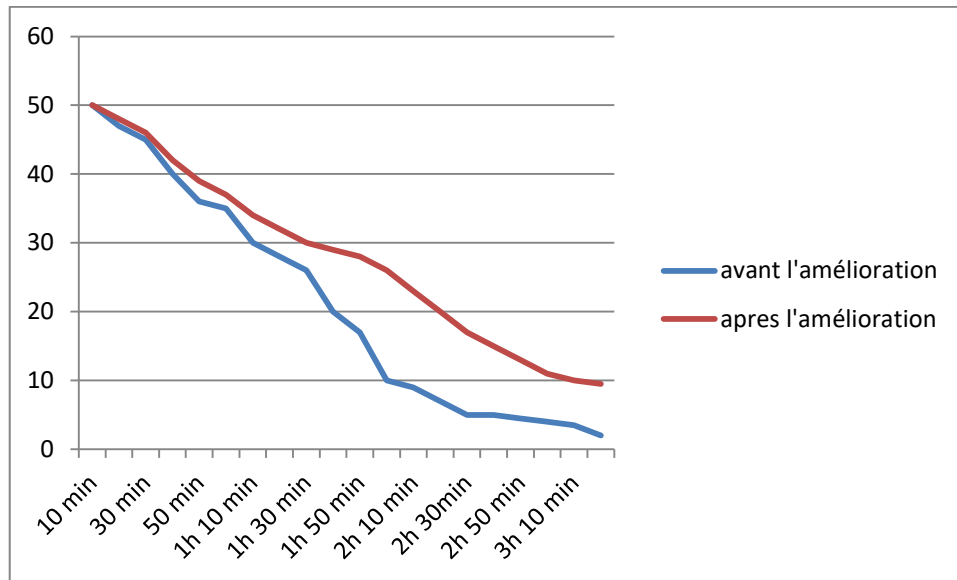


Figure V.3: Résultat de scénario 1.

Résultats et discussion :

La figure ci-dessus compare l'évolution de la consommation de la batterie d'un Smartphone sous deux approches (l'approche AVSCU et notre solution).

Ces résultats montrent clairement que la consommation de la batterie avec l'utilisation de l'approche AVSCU (Avant l'amélioration) est supérieure à celle de notre solution (Après l'amélioration), la différence commence après le 50%. Considérant que les qualités que les utilisateurs obtiennent lorsqu'ils regardent la vidéo à l'aide de notre solution sont: 720p, 480p, 360p à 1h10min et 480p, 360p, à 2h10min et 3h10min nous avons une seule qualité qui est 360p. Cette réduction de la qualité minimise l'utilisation de la charge, économisant ainsi la consommation de la batterie. Contrairement, l'approche AVSCU continue à recevoir la même qualité vidéo ce qui affaiblit la batterie rapidement.

En déduire qu'après avoir modifié les pondérations de la batterie, le facteur d'utilisateur maintenant est calculé en basant beaucoup plus sur la batterie que les autres paramètres et

donc cela réduira la qualité en fonction de la charge, économisant ainsi la consommation de la batterie.

V.3.2 Scénario 2

Le but de ce test est de voir la différence d'affectation des qualités vidéo pour le même terminal lorsqu'en utilisent l'approche AVSCU et notre solution. On a un utilisateur entraine de regarder une vidéo Youtube avec son ordinateur portable on mesurant la qualité en fonction de niveau de batterie disponible.

La figure suivante montre les résultats de ce test :

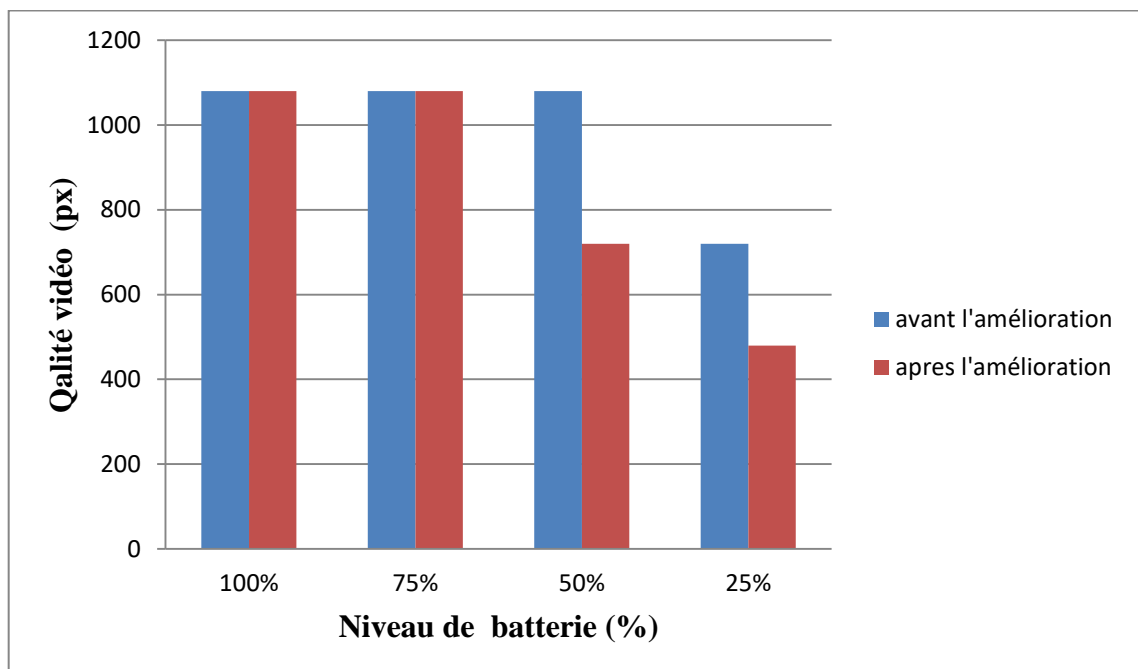


Figure V.4: Résultat de scénario 2.

Résultats et discussion :

D'après la figure ci-dessus nous avons noté que l'affectation des qualités vidéo avant l'amélioration ne donne pas une grande importance au niveau de batterie disponible, nous voyons qu'entre 100% et 25% le terminal continue à recevoir la même qualité vidéo (1080 p) jusqu'à qu'il arrive à 25%, la qualité reçue ici est 720p, on peut dire donc que l'approche AVSCU (avant l'amélioration) vise à maximiser les qualités vidéo même si au détriment de la batterie. Par contre, dans notre solution nous voyons que la qualité vidéo respecte le niveau de la batterie c'est-à-dire la qualité perçue est réduite si le niveau de batterie est faible et vice versa.

V.3.3 Scénario 3

Afin de savoir le feedback des clients nous avons réalisé ce scénario on amène un groupe des utilisateurs de 10 personnes. Chaque utilisateur a visionné la même vidéo Youtube avec leurs terminaux. Par la suite, chaque utilisateur évalue la qualité vidéo en donnant son note entre 1 et 5. Enfin, nous avons calculé le score moyen MOS.

Grâce à la mise en œuvre du test, les résultats suivants ont été obtenus:

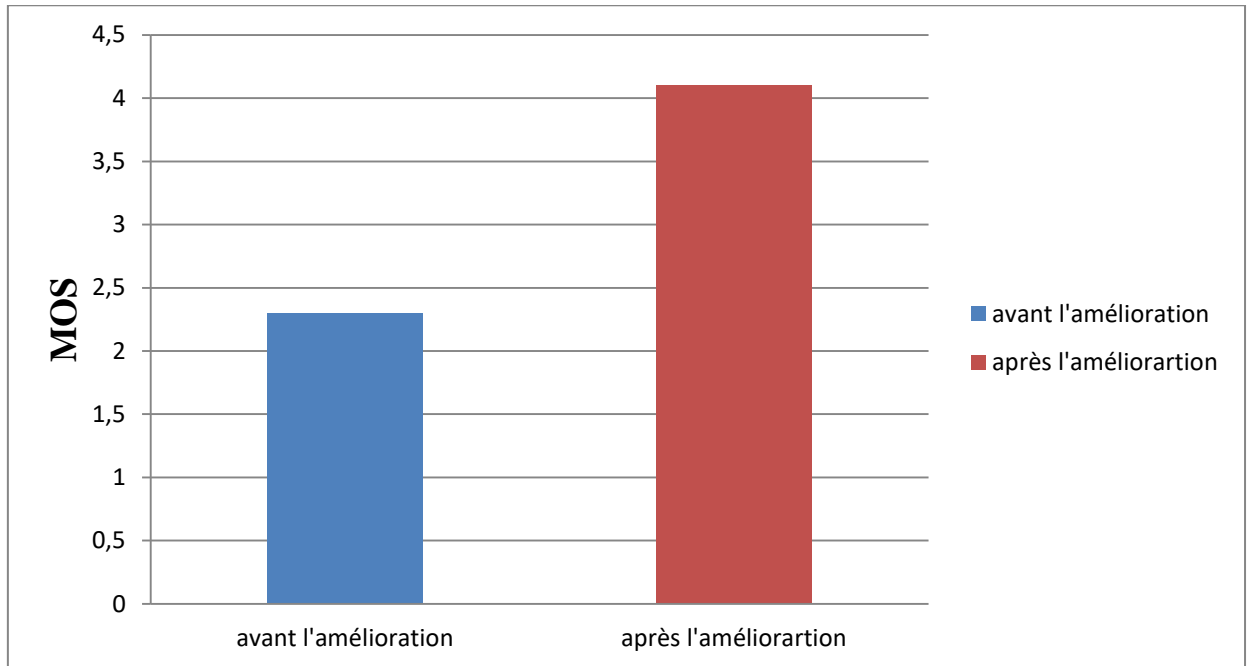


Figure V.5: Résultat de scénario 3.

Résultats et discussion :

De ce graphe, nous remarquons que la valeur donnée par les clients dans le premier test (avant l'amélioration) est de 2,3, ce qui représente la deuxième valeur (pauvre) et la valeur de MOS donnée dans le deuxième test (avec l'amélioration) est 4,1 ce qui est égal à le quatrième niveau de MOS (bon). Par conséquent, le résultat du test de notre solution est meilleur que Résultats des tests utilisant la méthode AVSCU (sans amélioration).

On peut en conclure que la plupart des utilisateurs ne sont pas intéressés par la qualité vidéo comme il est important de continuer à regarder des vidéos et d'économiser l'énergie de la batterie.

V.4 Conclusion

Dans ce chapitre nous avons implémenté notre approche pour le but d'améliorer la qualité d'expérience des services de vidéo streaming adaptatif dans une architecture MEC.

Tout d'abord nous avons présenté en détail l'environnement d'émulation de la solution ainsi que les différents outils d'implémentation. Puis nous avons discuté les résultats des scénarios qui ont prouvé l'efficacité de notre solution.

Conclusion générale

CONCLUSION GENERALE

Vu la révolution qu'a subi la recherche dans le domaine du streaming et l'augmentation continue du trafic vidéo sur l'internet, impose aux opérateurs de répondre efficacement aux phénomènes de congestion et d'améliorer ses services avec des mesures qui apparaissent pour satisfaire leurs clients. Dans ce rapport nous nous sommes intéressés en particulier d'améliorer la qualité d'expériences des services de vidéo streaming adaptatif dans une architecture MEC.

Pour atteindre cet objectif nous avons franchi deux étapes fondamentales.

Dans la première étape nous avons évalué l'approche AVSCU avec des tests de qualité d'expérience, les tests que nous avons réalisés sont basés sur les paramètres d'utilisateur tel que la batterie, le CPU, les données internet..., grâce aux résultats de ces tests nous avons déduire que l'utilisation de l'approche AVSCU nous permet de lire une vidéo avec moins de coupure, aussi d'économiser la consommation du forfait internet. En revanche, il put y avoir aussi des inconvénients en mettre en valeur le problème de batterie et de congestion.

Dans la deuxième étape on a essayé d'améliorer l'approche AVSCU.

Tout d'abord nous avons proposé de modifier le facteur de la batterie FB, pour donner plus d'importance à la batterie dans le calcul du UF, en suite nous avons proposé une formule probabiliste pour calculer le risque de la congestion.

L'efficacité des améliorations qu'on a réalisées ont été illustrée par des tests dans le dernier chapitre.

L'amélioration de la qualité de service est également un enjeu majeur, c'est pourquoi, comme perspective de ce travail, il serait intéressant de considérer les paramètres du réseau pour apporter de meilleures solutions.

Bibliographie

Bibliographie

- [1] P. Miniset *al.*, “Gestion de la Qualité de Service des flux Streaming dans les Réseaux 802. 11,” 2012.
- [2] C. Universitaire and B. Bouchaib, « Gestion de la QoS des flux vidéo dans un réseau hétérogène ».
- [3]<https://web.maths.unsw.edu.au/~lafaye/CCM/internet/rtcp.htm> (accessed Oct. 17, 2020).
- [4]www.techopedia.com/definition/4753/real-time-streaming-protocol-rtsp (accessed Oct. 17, 2020).
- [5] M.Charchar, “Amélioration de la qualité d’expérience des services de vidéo streaming adaptatif dans une architecture MEC, 2019.
- [6] A. Lavignotte, “Prise en compte de la qualité de l’expérience utilisateur au sein des protocoles de streaming HTTP adaptatifs,” 2014.
- [7]<https://web.maths.unsw.edu.au/~lafaye/CCM/internet/qos-qualite-de-service.htm> (accessed Sep. 25, 2020).
- [8] B. Errol, D. Dagbenonbakin, M. Présenté, and À. L. É. D. E. Technologie, “Analyse et conception de modèle de mesure de la qualité d’expérience perçue lors de la diffusion de la vidéo dans les réseaux sans-fil hétérogènes,” 2018.

- [9] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile Edge Computing: A Survey,” *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, 2018, doi: 10.1109/JIOT.2017.2750180.
- [10] www.bogotobogo.com/VideoStreaming/mpeg_dash.
- [11] G. Lefebvre “Optimisation Multi-Critères pour la Diffusion Vidéo au sein de l’Internet Media du Futur,” thèse de doctorat, Novembre 2017.
- [12] Y. Douga. “Amélioration de la performance de TCP dans les réseaux mobiles ad hoc“, thèse de doctorat, Novembre 2016.
- [13] H. Thalmensy, “Emulation de réseaux au niveau IP pour l’expérimentation de services et protocoles de communication. Application aux réseaux satellites,” 2007, [Online]. Available: <http://hal.archives-ouvertes.fr/tel-00197183/>.
- [14] “Ubuntu.com Competitive Analysis, Marketing Mix and Traffic - Alexa.” <https://www.alexa.com/siteinfo/ubuntu.com> (accessed Oct. 18, 2020).
- [15] DASH Industry Forum. For Promotion of MPEG-DASH 2013. Available: <http://dashif.org>
- [16] www.ffmpeg.org, consulté octobre 2020.
- [17] www.gpac.wp.imt.fr/mp4box, consulté octobre 2020.
- [18] <https://github.com/Dash-Industry-Forum/dash.js>, consulté octobre 2020.
- [19] www.httpd.apache.org/docs, consulté octobre 2020.
- [20] [www.https://documentation.online.net/fr/dedicatedserver/tutorials/network/configurewondershaper](https://documentation.online.net/fr/dedicatedserver/tutorials/network/configurewondershaper), consulté octobre 2020.

- [21] <https://docs.mitmproxy.org/stable/tools-mitmproxy>, consulté octobre 2020.
- [22] <https://doc.ubuntu-fr.org/cron>, consulté octobre 2020.
- [23] www.python.org/doc, consulté octobre 2020.
- [24] C. Ben Ameer, “Streaming Chiheb Ben Ameer TCP Protocol Optimization for HTTP Adaptive Streaming,” 2016.
- [25] “Plateformes de vidéo en streaming : Guide 2020 - Blog Cobra.” <https://blog.cobrason.com/2020/02/plateformes-de-video-en-streaming-guide-2020/> (accessed Dec. 11, 2020).
- [26] <https://www.cbnews.fr/etudes/image-pollution-streaming-video-occupe-606-du-trafic-global-internet-47109>

Annexe

Algorithme AVSCU (adaptation de vidéo streaming centralisé basé utilisateur) :

L'idée générale de cet algorithme est la suivante :

Nous mettons la station MEC dans la demande d'écoute

- ✓ Si la requête contient des paramètres utilisateur. Ensuite, nous calculons le facteur utilisateur en fonction de ces paramètres dans le tableau contenant tous les utilisateurs. Cette opération est décrite à l'étape 1.
- ✓ Si la requête contient une demande de fichier MPD, nous activerons la fonction de contrôle de congestion pour modifier les facteurs de l'utilisateur afin qu'aucune congestion ne se produise pendant que le facteur d'origine est corrigé. Sinon, nous le corrigerons. pas encore. Par conséquent, nous avons filtré la qualité des fichiers MPD et les avons envoyés aux utilisateurs. Cette opération est expliquée aux étapes 2 et 1.
- ✓ Si la demande comprend une évaluation, enregistrez-la dans le même tableau. L'étape 3 est requise pour cette évaluation.

La station MEC déclenche le mécanisme à une heure prédéfinie.

- ✓ La fonction de MOS, elle se fait si et seulement c'est l'heure est 00 :00. On justifie ce choix pour la raison de collecter les évaluations durant 24 heures. Cette opération est expliquée dans l'étape 3.

Dans cette section, sois :

- **Req** : la requête lors les échanges HTTP.
- **Table[]** : c'est une table qui contient des lignes, chaque ligne correspond à un utilisateur, les champs de la ligne sont :

[Adr_mac , Fu , Fu_m , T , Eval] , tel que :

- **Adr_mac** : adresse mac de l'utilisateur.

-
- **Fu** : c'est le facteur-utilisateur réel.
 - **Fu_m** : c'est le facteur utilisateur modifié par la fonction de contrôle de congestion, il est le facteur utilisé pour filtrer le fichier MPD.
 - **T** : c'est le temps enregistré lors des échanges de MPD pour la raison de vérifier si l'utilisateur est en train de regarder.
 - **Eval** : évaluation envoyée par l'utilisateur.
 - **Paramètres []** : ce sont les paramètres de l'utilisateur :
[Résolution,Taille,Batterie] , tel que :
 - **Résolution** : c'est la résolution de l'écran.
 - **Taille** : c'est la taille de l'écran.
 - **Batterie** : c'est le niveau de la batterie disponible.
 - **MPD** : le fichier MPD.
 - **RC** : le risque de la congestion.
 - **PBU** : Bande passante consommée par les utilisateurs.
 - **PBT** : Bande passante totale que l'enode B peut servir.
 - **V** : Valeur de dégradation de FU.
 - **T** : le temps.
 - **FM** : le facteur de MOS.

Var Req, Table [], Paramètres[] : chaîne de caractère ;

MPD : fichier ;

Fu, Fu_m, FM=0 : réel ;

Evaluation : entier ;

Début

Tant que (Vrai) faire //mettre la station MEC en écoute

Si(Trafic('requête')) alors

Req ← Trafic('requête') ; //obtenir la requête http

Adr_mac ← *Obtenir_adr_mac_source*(req) ; //Obtenir l'adresse mac de ce qui
 envoie la requête

Si(Req.contien('paramètres')) alors

Paramètres[] ← *Obtenir_parameters*(req,résolution,taille,batterie) ;

Fu ← *Calculer_fu*(*Paramètres*[],*FM*) ;

Maj(*Table*[],*Adr_mac*,*Fu*,*Fu_m*,*R*) ;

Sinon Si (*Req.contien*('mpd')) alors

Table[] ← *Contrôle_congestion*(*Table*[]) ;

MAJ(*Table*[],*Adr_mac*,*Obtenir_temps*()) ; //ajouter le temps à la ligne
 d'utilisateur

MPD ← *Obtenir_response*(*Req*) ;

Fu_m ← *Obtenir_fu_m*(*Table*[],*Adr_mac*) ;

MPD ← *MAJ*(*MPD*,*Fu_m*) ; // Filtrer les qualités de vidéo selon fu_m

Envoyer(*MPD*,*Adr_mac*) ;

Sinon Si(req.contien('évaluation')) alors

Evaluation ← *Obtenir_evaluation*(*Req*) ;

MAJ(*Table*[],*Adr_mac*,*Evaluation*) ; //ajouter l'évaluation à la ligne
 d'utilisateur

Fin Si ;

Sinon Si (*Heure*()='00 :00')

FM ← *Facteur_de_MOS*(*FM*,*Table*[]) ;

Fin Si ;

Fin Si ;

Fin Faire ;

Fin.
