

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار - عنابة

Année : 2019

Faculté: Sciences de l'Ingénierat
Département: Electronique

MEMOIRE

Présenté en vue de l'obtention du diplôme de : MASTER

Intitulé :
**Évaluation de la qualité vidéo
dans une transmission sans fil**

Domaine : Sciences et Technologie
Filière : Télécommunications
Spécialité : Réseaux et Télécommunications

Par :
Senoussi Mohamed

DEVANT Le JURY

Président : S.affifi	Prof	UBM Annaba
Directeur de mémoire : S.Harize	MCA	UBM Annaba
Examineur1 : M.boutalbi	MCB	UBM Annaba
Examineur2 : M.Benouaret	Prof	UBM Annaba

Remerciements

J'adresse en premier lieu ma reconnaissance à notre DIEU le tout puissant, de m'avoir donné la capacité, la patience d'aller jusqu'au bout du rêve.

J'adresse ma reconnaissance, ma gratitude à mon encadreur le professeur **Mme. Saliha Harize** de m'avoir fait bénéficier de ces compétences, ses qualités humaines et de sa disponibilité non seulement pour la réalisation de ce mémoire mais aussi durant tout le parcours de ma formation.

J'exprime mes sincères remerciements à mes parents qui m'ont enseigné la patience, la politesse, le sacrifice et qui ont toujours été là pour moi, Que le bon DIEU les bénisse d'avantage.

Un très grand merci à mes frères de me donner le courage pour terminer ce travail.

Sans oublier tous mes très chères amies : Seif, Islam, Ahmed, Khaled, et Nour, spécialement Islam Mansri et Mme. Sara Khalifa qui m'a beaucoup aidé et m'a encouragé pendant la période de la réalisation de mon projet.

Je n'oublie pas de dire un grand merci à toutes les personnes, tous les professeurs qui ont contribué de près et de loin à l'enrichissement de mon travail et à mon épanouissement intellectuel.

Dédicaces

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, ma mère.

A mon père, écolle de mon enfance, qui a été mon ombre durant toutes les années des études, et qui a veillé tout au long de sa vie à m'encourager, à me donner l'aide et à me protéger.

Que dieu les garde et les protège.

A mes frères Hamza, raouf, et mouad pour toutes leurs affections.

A mes amis, Hamza, Mohamed, Khalid, Zaki, Islam, Fares, Djalel, Anis, Wail, Hamza Chakour, Brahim, Marwan, et Houcine ...

Un grand merci à Mme. Lynda qui m'a poussé à me relever.

A tous ceux qui sont chères.

A tous ceux qui m'aiment.

A tous ceux que j'aime.

Résumé :

La quantité de données échangée sur les réseaux est en croissance fulgurante et les fichiers vidéo sont de plus en plus volumineux à cause des exigences des utilisateurs en termes de qualité. La compression est alors l'unique solution, que ce soit pour le stockage ou pour la transmission. D'autre part, les réseaux sans fil sont connus pour être hostiles aux transmissions notamment la vidéo. En effet, ces canaux de transmission introduisent inévitablement des bruits et des pertes de paquets, ce qui va dégrader la qualité de service (QoS) et/ou la qualité d'expérience (QoE).

Ce travail s'inscrit dans ce contexte et a pour objectif principal l'évaluation de l'impact des pertes de paquets (PLR) sur la qualité des vidéos compressées par le codec HEVC à différents taux et diverses configurations lors de leur transmission à travers un réseau ad-hoc sans fil. Les logiciels FFmpeg, l'émulateur de réseaux NETEM et le protocole RTP ont été utilisés dans la transmission des vidéos dont la qualité a été évaluée en utilisant le PSNR.

Mots clés : HEVC, FFmpeg, NETEM, RTP, PLR, PSNR, QoE, QoS

Abstract :

The amount of exchanged data over the networks is in a tremendous growth and video files are increasingly huge because of the users requirements in terms of quality. Compression is therefore, the unique solution, either for storage purposes or for transmission. On the other hand, wireless networks are known for being hostile to transmission and especially the video. Indeed, these channels introduce inevitably noise and packet losses (PLR) which will degrade the quality of service (QoS) and/or the quality of experience (QoE).

This work is carried in this context and has as main goal the evaluation of packet losses impact on the quality of HEVC compressed videos at different rates and various configurations when they are transmitted on a ad-hoc wireless network. FFmpeg, network emulator NETEM and the RTP protocol were used in the video transmission which quality has been evaluated using the PSNR.

Key words: HEVC, FFmpeg, NETEM, RTP, PLR, PSNR, QoE, QoS

الملخص:

تنمو كمية البيانات التي يتم تبادلها على الشبكات بشكل سريع وأصبحت ملفات الفيديو أكبر بسبب متطلبات المستخدمين من حيث الجودة. إذا الحل الوحيد هو عملية الضغط، سواء للتخزين أو للإرسال. من ناحية أخرى، من المعروف أن الشبكات اللاسلكية معادية للإرسال بما في ذلك الفيديو. في الواقع، تؤدي قنوات الإرسال هذه حتماً إلى خلق تشويش (bruit) وفقدان الحزم (Perte des paquets)، مما يؤدي إلى انخفاض جودة الخدمة (QoS) و / أو جودة التجربة (QoE).

هذا العمل جزء من هذا السياق وهدفه الرئيسي هو تقييم تأثير فقد الحزم (PLR) على جودة مقاطع الفيديو المضغوطة بواسطة برنامج الترميز HEVC بمعدلات مختلفة وتكوينات مختلفة عند إرسالها عبر الشبكة اللاسلكية ad-hoc.

البرامج FFmpeg، ومحاكي الشبكة NETEM، والبروتوكول RTP، تم استخدامهم في نقل مقاطع الفيديو التي تم تقييم جودتها باستخدام PSNR.

الكلمات المفتاحية: جودة الخدمة (QoS)، جودة التجربة (QoE)، HEVC، FFmpeg، NETEM، RTP، PLR،

PSNR.

LISTE DES ACRONYMES

A

AI: All Intra

AVC: Advanced Video Coding

D

DB: decibel

DDR: Double Data Rate

DHCP : Dynamic Host Configuration Protocol

DVB : Digital Video Broadcasting

F

FFMPEG: Fast Forward MPEG

FPS: Frames per Second

G

GNS3 : Graphical Network Simulator-3

GOP : Group of Picture

H

HEVC: High Efficiency Video Coding

HM: HEVC Test Model

I

IEC: International Electrotechnical Commission

IETF : Internet Engineering Task Force

IOS : Internetwork Operating System

ISO: International Standardization Organization

IP: Internet Protocol

IPv4: Internet Protocol Version 4

J

JPEG: Joint Photographic Experts Group

L

LDP: Low Delay P

LTS : Long-Term Support

LZW : Lempel-Ziv-Welch

M

MANET : Mobil Ad hoc Network

MOS: Mean Opinion Score

MPEG: Moving Picture Experts Group

MSE: Mean Squared Error

N

NETEM :

N_F : Nombre de Frame

NS2 : Network Simulator 2

O

OMNeT++ : Objective Modular Network Testbed in C++

OS: Operating Systems

OSI : Open Systems Interconnection

P

PC : Personal Computer

PNG: Portable Network Graphic

PLR: Packet Loss Rate

PSNR: Peak Signal to Noise Ratio

Q

QoS : Quality of Service

QP : Quantization Parameter

QDISC : Queuing Discipline

R

RA: Random Access

RAR: Roshal Archive

RFC : Request for Comments

RTP : Real Time Protocol

S

SDP: Session Description Protocol

SSIM: structural similarity

T

TC: Traffic Control

TCP/IP : Transmission Control Protocol

U

UDP : User Datagram Protocol

UHD: Ultra High Definition

UIT-T : International Telecommunication Union for Telecommunications

V

VCEG: Vidéo Coding Experts Group

VQMT: Video Quality Measurement Tool

W

WAN : Wide Area Network

WAV: Waveform Audio

LISTE DES FIGURES

Chapitre 1

Figure 1.1 Numérisation du signal vidéo	2
Figure 1.2 Echantillonnage couleur	3
Figure 1.3 Schéma de principe d'un Codec (AV1)	5
Figure 1.4 processus d'encodage et décodage de H.264.....	5
Figure 1.5 processus d'encodage et décodage de H.265.....	6

Chapitre 2

Figure 2.1 Diagramme du modèle OSI.....	7
Figure 2.2 Diagramme du modèle TCP/IP.....	8
Figure 2.3 Modèle TCP/IP et OSI.....	8
Figure 2.4 Format du Paquet IP.....	10
Figure 2.5 Les Modes de transmission.....	11
Figure 2.6 Exemple d'une topologie Ad-hoc.....	11

Chapitre 3

Figure 3.1 Commande ffmpeg pour connaitre la version installée.....	16
Figure 3.2 Paramètres du réseau sans fil utilisé.....	19
Figure 3.3 Fichier SDP.....	20
Figure 3.4 Commande ffmpeg pour la réception du fichier	21
Figure 3.5 Commande ffmpeg pour la transmission	21
Figure 3.6 Indication ffmpeg sur la fin de la transmission	21
Figure 3.7 Indication ffmpeg sur la réception	22
Figure 3.8 Les fichiers au niveau client	22
Figure 3.9 Commande NETEM pour un PLR 1%	23
Figure 3.10 Pertes des paquets au niveau de la réception	23
Figure 3.11 Taille des fichiers transmis et reçus avec différents PLR (vidéo BQsquare).....	29
Figure 3.12 Taille des fichiers transmis et reçus avec différents PLR (vidéo Kimono).....	30

Figure 3.13 Taille des fichiers transmis et reçus avec différents PLR (vidéo RaceHorses)...	31
Figure 3.14 Pertes absolues en fonction de QP et PLR (a) (Kimono, AI), (b) (BQsquare, LD P), (c) (RaceHorses, LD P)	32
Figure 3.15 Pertes absolues en fonction de QP et PLR (a) (BQsquare, AI), (b)(Race Horses, AI), (c) (Kimono, LD P), (d) (BQsquare, RA), (e) (Kimono, RA), (f) (Race Horses, RA)	34
Figure 3.16 Variation du PSNR en fonction du PLR(a) (BQsquare, AI), (b) (Kimono, AI), (c) (RaceHorses, AI), (d) (BQsquare, LDP), (e) (BQsquare, RA), (f) (Kimono, RA),(g) (RaceHorses, RA), (h) ((BQsquare, Kimono, RaceHorses) non compressés)	37
Figure 3.17 Variation du PSNR en fonction de QP pour différentes configurations	37
Figure 3.18 Effet de PLR sur la vidéo (Kimono, Low Delay P, QP37)	40
Figure 3.19 Effet de PLR sur la vidéo (BQsquare, ALL INTRA, QP22)	40
Figure 3.20 Effet de PLR sur la vidéo (RaceHorses, Random Access, QP32)	40

LISTE DES TABLEAUX

Chapitre 3

Tableau 3.1 Caractéristiques du matériel utilisé	15
Tableau 3.2 Caractéristiques des vidéos utilisées	19
Tableau 3.3 Résultats du test de la transmission	22
Tableau 3.4 Résultat de la transmission de la vidéo BQsquare en mode ALL Intra	24
Tableau 3.5 Résultat de la transmission de la vidéo Kimono en mode ALL Intra	24
Tableau 3.6 Résultat de la transmission de la vidéo RaceHorses en mode ALL Intra	25
Tableau 3.7 Résultat de la transmission de la vidéo BQsquare en mode Low Delay P	25
Tableau 3.8 Résultat de la transmission de la vidéo Kimono en mode Low Delay P	26
Tableau 3.9 Résultat de la transmission de la vidéo RaceHorses en mode Low Delay P	26
Tableau 3.10 Résultat de la transmission de la vidéo BQsquare en mode Random Access ...	27
Tableau 3.11 Résultat de la transmission de la vidéo Kimono en mode Random Access	27
Tableau 3.12 Résultat de la transmission de la vidéo RaceHorses en mode Random Access..	28
Tableau 3.13 Vérification du caractère aléatoire de la perte de paquets (Kimono, AI)	38
Tableau 3.14 (Kimono, All Intra) résultats PSNR avec différents outils	39
Tableau 3.15 (RaceHorses, LD P) résultats PSNR avec différents outils	39

Table des matières

Remerciement	I
Dédicaces	II
Résumé	III
Liste des Acronymes	V
Liste des figures	VIII
Liste des tableaux	X
Introduction générale	1
Chapitre 1 : Vidéo numérique	2
1.1 Introduction	2
1.2 Vidéo numérique	2
1.3 Numérisation du signal vidéo composite	2
1.4 Les formats vidéo	3
1.5 La compression vidéo	3
1.5.1 Compression sans perte	3
1.5.2 Compression avec perte	4
1.6 Evaluation de la qualité des vidéos compressées avec perte	4
1.7 Les codecs vidéo	4
1.8 Les normes de compression vidéo	5
1.8.1 H.264/MPEG-4/ AVC	5
1.8.2 H.265/HEVC	6
1.8.3 Comparaisons entre H.264 et H.265	6
1.9 Conclusion	6
Chapitre 2 : Réseaux, simulateurs et émulateurs de réseaux	7
2.1 Introduction	7
2.2 Modèles de référence	7
2.2.1 Le modèle de référence OSI	7
2.2.2 Le modèle de référence TCP/IP	8
2.2.3 Modèles OSI et TCP/IP	8

2.3 Les principaux protocoles utilisés en transmission vidéo	9
2.3.1 Le protocole TCP	9
2.3.2 Le protocole UDP	9
2.3.3 Le protocole RTP	9
2.3.4 Le protocole IP	10
2.4 Modes de transmission	10
2.4.1 Mode Unicast	10
2.4.2 Mode Broadcast	10
2.4.3 Mode Multicast	11
2.5 Les réseaux mobiles ad hoc	11
2.5.1 Les caractéristiques des réseaux ad-hoc	12
2.5.1.1 Une topologie dynamique	12
2.5.1.2 Une bande passante limitée	12
2.6 Hostilités du canal de transmission	12
2.6.1 Le délai	12
2.6.2 La latence	12
2.6.3 La gigue (jitter)	12
2.6.3.1 La relation entre la gigue et la latence	12
2.6.4 Perte de paquet	12
2.6.5 Le débit	13
2.7 Simulateurs et émulateurs de réseaux	13
2.7.1 Définition, différences	13
2.7.1.1 Simulateur de réseau	13
2.7.1.2 Emulateur de réseau	13
2.7.1.3 Différences entre simulateur et émulateur de réseau	13
2.7.2 Quelques simulateurs et émulateurs	13
2.7.2.1 NS2	13
2.7.2.2 OMNeT++	14
2.7.2.3 NETEM	14
2.7.2.4 GNS3	14
2.8 Conclusion	14
Chapitre 3 : Simulations, résultats et analyses	15
3.1 Introduction	15
3.2 Logiciel et matériel utilisés	15
3.2.1 Matériel	15
3.2.2 Logiciels	16
3.2.2.1 Installation de FFMPEG	16
3.2.2.2 Quelques commandes FFMPEG	16
3.2.2.3 Installation de NETEM	17

3.2.2.4 Quelques commandes NETEM	17
3.3 Simulations	18
3.3.1 Configuration du réseau	18
3.3.2 Le protocole utilisé	19
3.3.3 Les vidéos utilisées et leurs caractéristiques	19
3.3.4 Procédure de simulation	20
3.3.4.1 Test d'une transmission fiable PLR=0%	20
3.3.4.2 générer une perte de paquet	23
3.4 Résultats	23
3.4.1 En mode All Intra	24
3.4.2 En mode Low Delay P	25
3.4.3 En mode Random Access	27
3.5 Analyses et discussions	28
3.5.1 Impact du PLR sur la taille du Bitstream	28
3.5.2 Impact du PLR sur la taille perdue (absolue)	32
3.5.3 Etude de l'impact du PLR sur la qualité des vidéos	35
3.5.4 Vérification du caractère aléatoire des pertes de paquets	38
3.5.5 Vérification des valeurs des PSNR	39
3.5.6 L'effet de la perte de paquets sur la vidéo	39
3.6 Conclusion et perspectives	41
Conclusion et perspectives	42
Bibliographie	43

Introduction Générale & Problématique

La vidéo est devenue la donnée la plus utilisée et la plus échangée sur les réseaux. On la retrouve à la télévision, sur les PC, les smartphones, les tablettes ...etc. L'exigence des consommateurs en termes de qualité a mené à utiliser des résolutions de plus en plus importantes. Cependant, UHD «Ultra High Definition», 4K ou 8K conduit à des fichiers vidéo de tailles énormes qui nécessitent une bande passante large pour leur transmission. L'unique solution, dans ce cas, est la compression. Mais, réduire le bitrate entraîne inévitablement une perte de qualité. C'est la raison pour laquelle les scientifiques et les chercheurs ne cessent de développer des algorithmes conduisant au meilleur compromis possible entre le débit et la qualité. C'est le cas des dernières normes H.264/MPEG-04/AVC et H.265/HEVC.

Les vidéos compressées sont plus fragiles en raison de la perte d'information, notamment lorsqu'elles sont transportées sur des canaux hostiles qui introduisent des perturbations telles le retard, la perte des paquets, la gigue... etc. la perte de paquet est plus prononcée dans les réseaux sans fil, en particulier les réseaux Ad-hoc qui sont caractérisés par une bande passante limitée et une topologie dynamique. La perte de paquets se produit lorsqu'un ou plusieurs paquets de données transitant sur un réseau n'atteignent leur destination. La perte de paquets est due soit à des erreurs de transmission de données, généralement sur des réseaux sans fil, soit à un encombrement du réseau. C'est l'un des problèmes les plus importants surtout dans les applications multimédia telles que le streaming vidéo qui diffuse la vidéo en temps réel et principalement sur des réseaux sans fil et où la QoE (Quality of Experience) est l'objectif principal.

Ce travail consiste donc à transmettre des vidéos de différentes résolutions, frame rates et durées, compressées par le HEVC à différents taux et différentes configurations, à travers un réseau ad-hoc sans fil. L'objectif est, d'une part, l'évaluation de l'impact du taux des pertes de paquets, générés par l'émulateur de réseaux NETEM, sur la qualité par la mesure du PSNR. D'autre part, les frames les plus sensibles doivent être déterminés.

Ce mémoire comprend trois chapitres. Dans le premier chapitre, des rappels sur la vidéo, ses caractéristiques, les méthodes de compression et les métriques d'évaluation de la qualité ont été présentés. Le deuxième chapitre présente les notions fondamentales sur les réseaux, les protocoles, les hostilités des canaux de transmission ainsi que les simulateurs et émulateurs réseaux les plus populaires. Le troisième chapitre expose la méthodologie de travail, récapitule les résultats obtenus ainsi que leurs analyses.

Chapitre 1 : Vidéo numérique

1.1 Introduction :

De nos jours, le monde connaît un grand développement dans le domaine de la vidéo numérique et ce à partir la vidéo analogique.

En outre, comme tous les signaux analogiques doivent passer par une conversion numérique pour être traités, stockés ou transmis à travers des chaînes numériques, le signal vidéo doit inévitablement subir une numérisation.

Mais cette procédure résulte en des fichiers énormes et la transmission de la vidéo numérique sur n'importe quel type de canal (streaming, DVB...) exige obligatoirement une compression vidéo généralement avec perte.

Dans ce chapitre on verra les types de compression de vidéos avec les normes et les codecs utilisés ainsi que l'outil de mesure la qualité d'une vidéo.

1.2 Vidéo numérique :

La vidéo numérique est une succession d'images échantillonnées spatialement et temporellement. L'œil humain a comme caractéristique de distinguer 20 images/seconde, donc si plus de 20 images numériques (généralement 25 ou 30 images/seconde) sont affichées, l'œil humain voit une image animée.

1.3 Numérisation du signal vidéo composite :

Un signal vidéo composite couleur est composé de la luminance Y et deux composantes chrominance Cr et Cb. Ces trois composants doivent être échantillonnés en respectant la condition de Shannon ($F_e \geq 2 \times F_{ma}$). La largeur de bande de Y est de l'ordre 6 MHz, donc la fréquence d'échantillonnage adaptée pour Y est 13.5 MHz. Pour Cr et Cb, nous pouvons prendre la fréquence la plus faible, soit 6,75 MHz pour réduire la redondance.

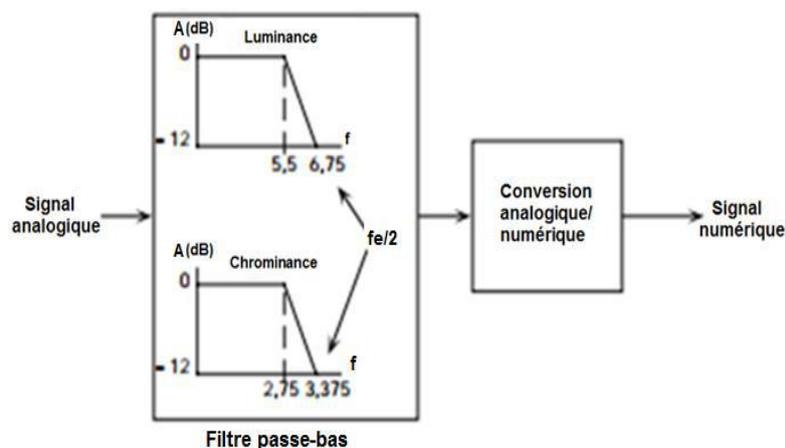


Figure 1.1 Numérisation du signal vidéo

1.4 Les formats vidéo :

Dans la vidéo numérique, nous avons plusieurs formats d'échantillonnage :

Le format 4:4:4 : YCrCb sont échantillonnés avec la même cadence 13.5 MHz.

Le format 4:2:2 : Y est échantillonné à 13.5 MHz alors Cr Cb sont chacun échantillonné à 6.75 MHz.

Le format 4:1:1 : Y est échantillonné à 13.5 MHz et Cr Cb sont chacun échantillonné à 3.375 MHz ($\frac{1}{4}$ de la luminance).

Le format 4:2:0 : Y est échantillonné à 13,5 MHz et les chrominances à 6,75 Mhz mais une ligne sur deux.

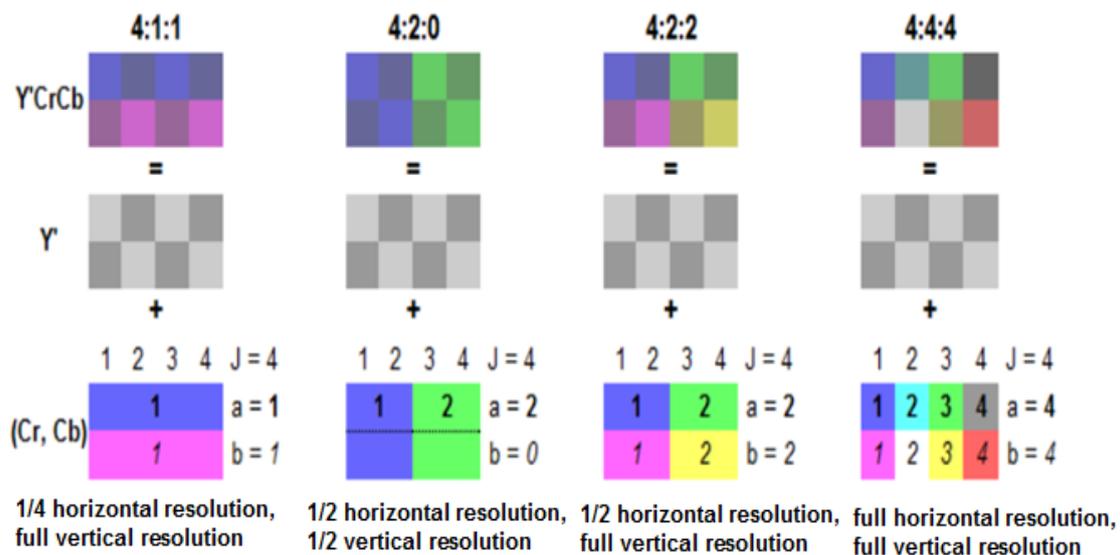


Figure 1.2 Echantillonnage couleur

1.5 La compression vidéo :

La vidéo numérique est certainement l'information la plus encombrante en termes de volume de données à stocker et de la bande passante de transmission. La solution est de compresser la vidéo avant de la stocker ou la transmettre. Il existe deux types de compression :

1.5.1 Compression sans pertes :

Est un algorithme de compression de données qui permet de reconstruire après décompression un fichier identique à l'origine, donc aucune perte de données sur l'information d'origine.

Exemples : codage Huffman, codage arithmétique, codage par dictionnaire LZW LZ77 LZ78....

Application: WAV, BMP, RAR, ZIP, PNG...

1.5.2 Compression avec pertes :

La compression avec pertes produit des données différentes des données originales.

D'autre part, avec la compression avec pertes nous pouvons atteindre un taux élevé de compression de données qu'avec la compression sans pertes mais au détriment d'une certaine perte de qualité.

1.6 Evaluation de la qualité des vidéos compressées avec pertes :

Pour évaluer la qualité d'une vidéo compressée avec pertes, on utilise habituellement des métriques basées sur des formules mathématiques telles que le PSNR, le MSE ou le SSIM et aussi des métriques subjectives basées sur la perception humaine (MOS).

Dans ce travail, on a utilisé principalement le PSNR dont la définition est :

"Est un terme technique qui désigne le rapport entre la puissance maximale possible d'un signal et la puissance de corruption du bruit, le PSNR est généralement exprimé en décibels. [1]"

Le PSNR se définit le plus facilement via l'erreur quadratique moyenne (MSE), est définie comme suit : [1]

$$MSE = \frac{1}{M \times N} \sum_{m=1}^M \sum_{n=1}^N (I(m, n) - \hat{I}(m, n))^2$$

Où $(M \times N)$ la taille de l'image

I et \hat{I} sont les amplitudes des pixels sur les images originale et dégradée (pour \hat{I} p).

Le PSNR (en dB) est défini comme suit : [1]

$$PSNR = 10 \log_{10} \frac{I_{max}^2}{MSE}$$

I_{max} désigne la luminance maximale possible

Une valeur de PSNR inférieure à **30 dB** traduit généralement une image présentant des dégradations perceptibles ; le PSNR est l'évaluation la plus couramment utilisée. [2]

1.7 Les codecs vidéo :

Un codec est l'implémentation matérielle ou logicielle qui met en œuvre une norme. [3]

" Codec " est une concaténation de "codeur" et de "décodeur", un périphérique qui compresse s'appelle généralement un codeur et celui qui décompresse est un décodeur. [4]

Exemple : X264, X265, VP9, AV1...

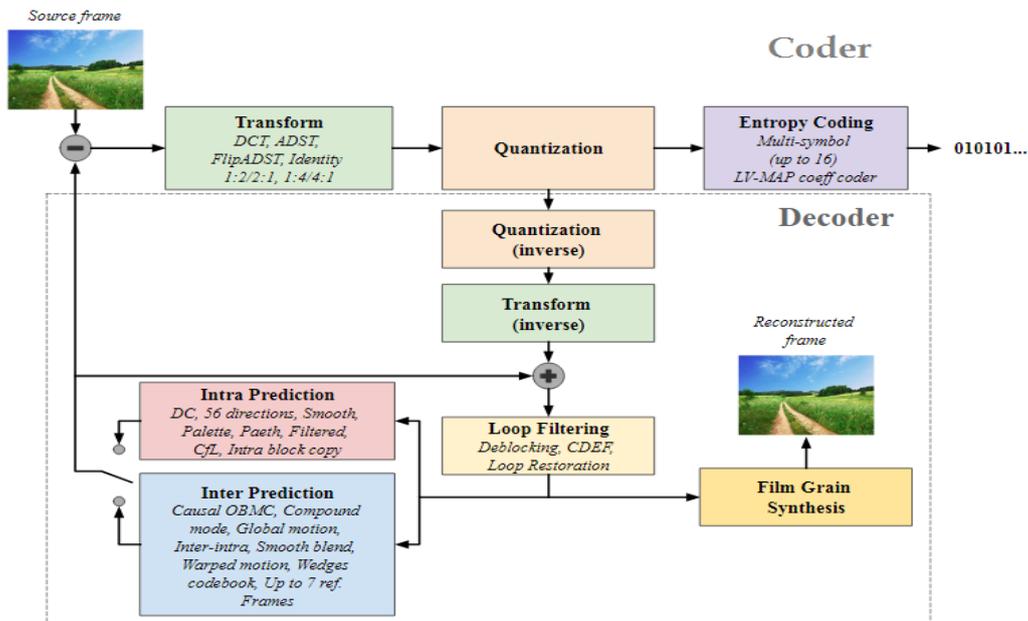


Figure 1.3 Schéma de principe d'un Codec (AV1)

1.8 Les normes de compression vidéo :

La norme décrit un procédé, pouvant définir un format conteneur ou un format de compression. [3]

Il y a plusieurs types de normes comme : JPEG, MPEG-1, MPEG-2, H.261, H.263, H.264/MPEG-4/ AVC, H.265/HEVC

Nous expliquons les deux dernières technologies :

1.8.1 H.264/MPEG-4/AVC :

H.264 ou MPEG-4, Partie 10, Codage vidéo avancé (MPEG-4 AVC) est une norme de compression vidéo basée sur la compensation de mouvement orientée par blocs. Depuis 2014, il s'agit de l'un des formats les plus utilisés pour l'enregistrement, la compression et la distribution de contenu vidéo. [1]

Il prend en charge les résolutions allant jusqu'à 8192 × 4320, y compris 8K UHD. [5]

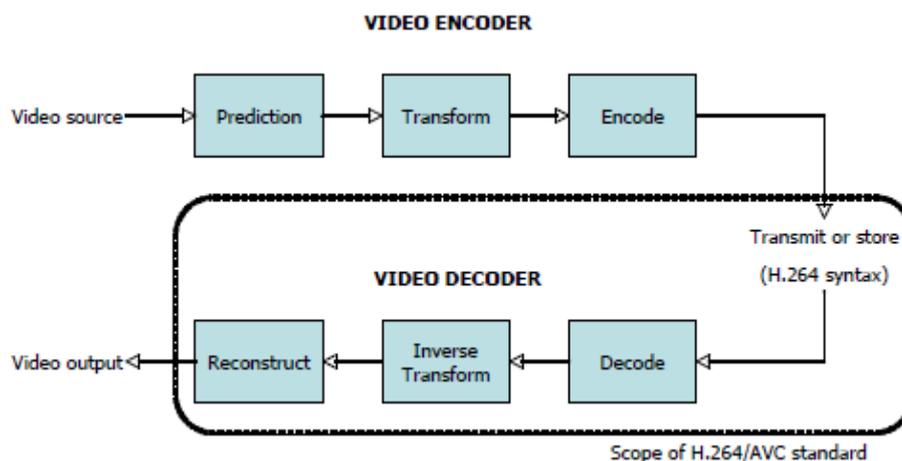


Figure 1.4 processus d'encodage et décodage de H.264

1.8.2 H.265/HEVC :

H.265 ou HEVC (High Efficiency Vidéo Coding) est une norme de compression vidéo élaborée par ISO/IEC 23008-2 et UIT-T H.265 et publiée le 13 avril 2013. Elle est développée conjointement par les groupes Vidéo Coding Experts Group (VCEG) et Moving Picture Experts Group et doit succéder au H.264 (ISO/IEC 14496-10 et UIT-T H.264). Ses applications concernent aussi la compression des vidéos en très haute définition (UHD, 2K, 4K, 8K...). [6]

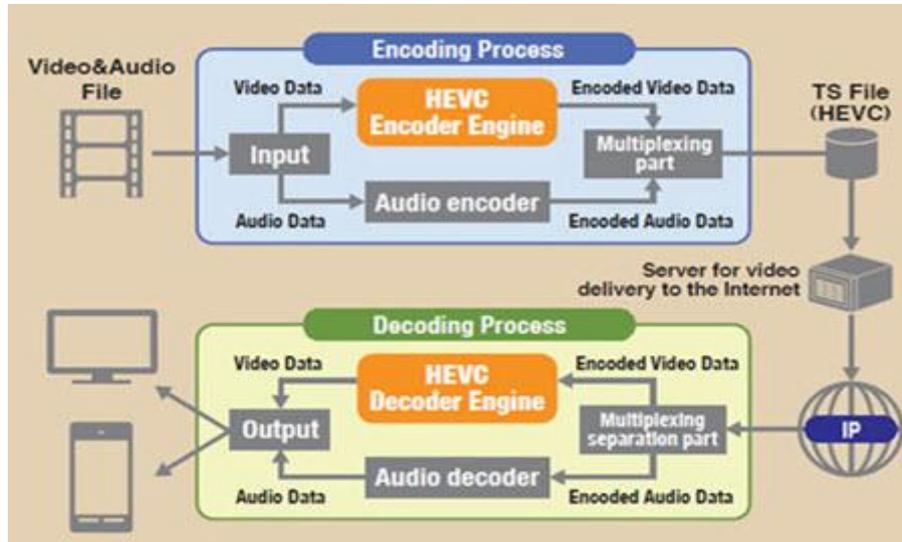


Figure 1.5 processus d'encodage et décodage de H.265

1.8.3 Comparaisons entre H.264 et H.265 :

HEVC est la norme la plus récente et a un grand succès par rapport au H.264. Elle est adaptée pour des résolutions vidéo jusqu'au UHD, qui peut atteindre jusqu'à 8K (8192*4320) pixels alors que la H.264 peut atteindre jusqu'à 4K (4,096*2,304). [7]

Pour un même débit, la norme HEVC produit une meilleure qualité que la H.264 et la HEVC promet d'énormes économies de bande passante d'environ 50% par rapport au H.264 avec la même qualité utilisée pour les deux. [7]

1.9 Conclusion :

Dans ce chapitre, des rappels succincts sur l'évolution de la vidéo numérique, la numérisation du signal vidéo couleur en composantes YCrCb ainsi que les principaux formats utilisés ont été donnés.

Nous avons vu aussi les deux types de compression vidéo avec et sans pertes et la définition de la métrique principale de mesure de la qualité (PSN). Nous avons également parlé des codecs et des normes de compression vidéo les plus récentes (H.264 et H.265) et la différence entre elles.

Chapitre 2 : Réseaux, simulateurs et émulateurs de réseaux

2.1 Introduction :

Dans ce chapitre nous allons rappeler les deux modèles de communications avec leurs couches et protocoles. Des précisions sur les protocoles de transport seront fournies vu leur importance dans la vie quotidienne. Nous parlerons aussi des modes de transmission, des réseaux Ad-hoc et des métriques de la qualité de service. Ce chapitre continue avec des aperçus sur les simulateurs et les émulateurs de réseaux.

2.2 Modèles de référence :

2.2.1 Le modèle de référence OSI :

Le modèle OSI (Open Systems Interconnection) est une norme de communication, en réseau, de tous les systèmes informatiques. C'est un modèle de communications entre ordinateurs proposé par l'ISO (Organisation Internationale de Normalisation) qui décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions. [8]

Le modèle OSI est un modèle qui comporte 7 couches présenté dans la figure 2.1.

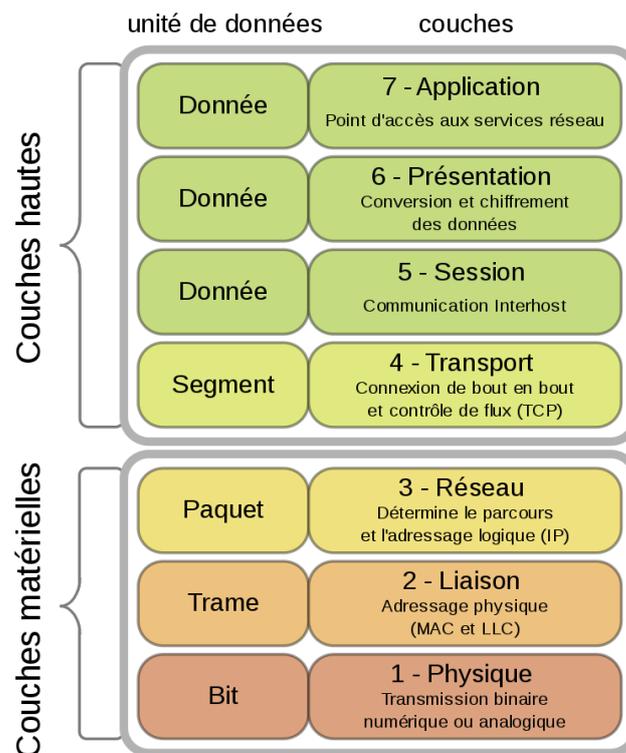


Figure 2.1 Diagramme du modèle OSI

2.2.2 Le modèle de référence TCP/IP :

Le modèle TCP/IP (Transmission Control Protocol/Internet Protocol) appelé aussi modèle Internet date de 1976. Il a été stabilisé bien avant la publication du modèle OSI en 1984. Il présente aussi une approche modulaire (utilisation de couches) mais en contient uniquement quatre : Application, Transport, Internet, Accès réseau. [8]

Aujourd'hui, c'est le modèle TCP/IP, plus souple, qui l'emporte sur le marché.

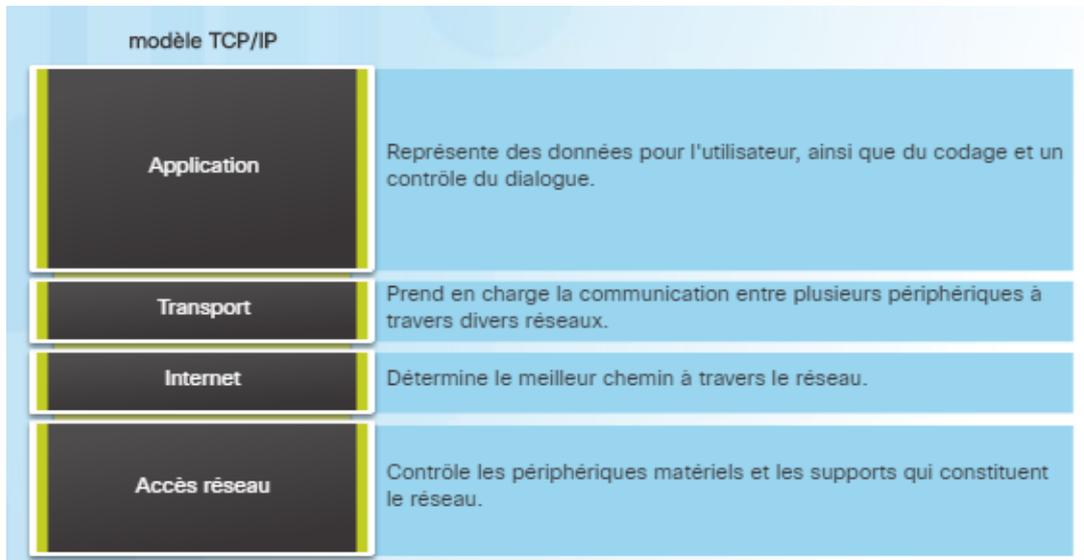


Figure 2.2 Diagramme du modèle TCP/IP

2.2.3 Modèles OSI et TCP/IP :

On trouve souvent des correspondances entre le modèle OSI et TCP/IP comme indiqué dans la figure 2.3

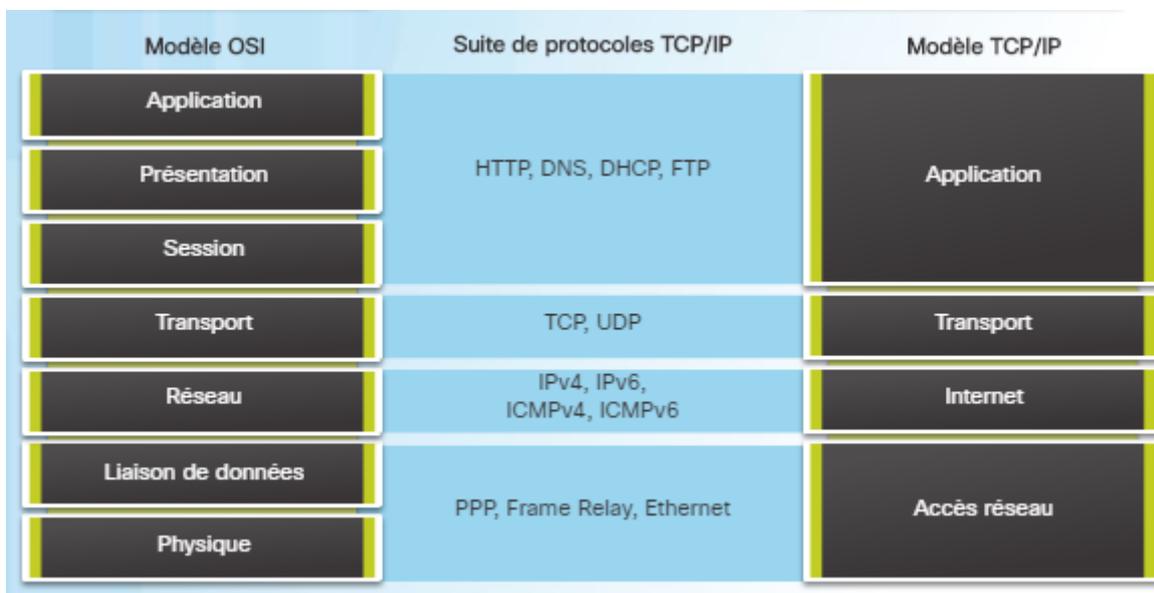


Figure 2.3 Modèle TCP/IP et OSI

2.3 Les principaux protocoles utilisés en transmission vidéo :

2.3.1 Le protocole TCP :

Le transport TCP «Transmission Control Protocol» revient à envoyer des paquets qui sont suivis de la source à la destination. Si la commande à expédier est divisée en plusieurs colis, un client peut vérifier en ligne l'ordre des livraisons.

Avec le protocole TCP, les trois fonctions de fiabilité de base sont :

- Numérotation et suivi des segments de données transmis à un hôte donné à partir d'une application spécifique
- Accusé de réception des données reçues
- Retransmission des données pour lesquelles aucun accusé de réception n'a été reçu, après un certain temps

TCP est incompatible avec le multimédia temps réel et les jeux en réseau. Plusieurs champs dans l'en-tête TCP sont indispensables (dont numéro de port et numéro de séquence) et devront se retrouver dans les protocoles de substitution UDP (User Datagram Protocol) et RTP (Real Time Protocol).

2.3.2 Le protocole UDP

UDP est un protocole de transport du modèle OSI utilisé souvent en streaming vidéo. Contrairement à TCP, UDP a très peu d'options à gérer et l'en-tête des paquets UDP est beaucoup plus petit, ce qui contribue à rendre son traitement plus rapide. Le transport est en mode non connecté est donc non fiable et sans garantie de séquençement des datagrammes. Il n'y a pas de contrôle de reprise sur erreur, mais UDP indique qu'il y a eu erreur. Il apporte la possibilité de distinguer plusieurs destinations sur un même ordinateur.

Un programme d'application qui utilise UDP doit gérer les problèmes de fiabilité, pertes de messages, duplications, retards, déséquençement et perte de connectivité. Chaque message UDP est appelé datagramme (datagramme UDP). Du point de vue conceptuel, le datagramme comporte deux parties : un en-tête fixe (8 octets) et une zone de données. **[9][10]**

2.3.3 Le protocole RTP :

Le protocole de transport en temps réel RTP «Real Time Protocol » est un protocole de réseau permettant de diffuser de l'audio et de la vidéo sur des réseaux IP. RTP est utilisé dans les systèmes de communication et de divertissement qui impliquent des médias de diffusion en continu, tels que la téléphonie, vidéoconférence applications... etc.

Le protocole RTP s'exécute généralement sur le protocole UDP (User Datagram Protocol). Le protocole RTP est utilisé conjointement avec le protocole de contrôle RTP (RTCP). Tandis que le protocole RTP achemine les flux multimédia (audio et vidéo, par exemple), le protocole RTCP est utilisé pour surveiller les statistiques de transmission et la qualité de service (QoS) et facilite la synchronisation de plusieurs flux. [11]

2.3.4 Le protocole IP :

Le protocole IP «Internet Protocol» a été conçu pour réaliser l'interconnexion de réseaux et permettre ainsi les communications entre systèmes. Il fait partie de la couche Internet de la suite des protocoles TCP/IP.

Le protocole Internet est responsable de l'adressage des interfaces hôtes, de l'encapsulation des données dans des datagrammes et du routage des datagrammes d'une interface hôte source vers une interface hôte de destination sur un ou plusieurs réseaux IP. À ces fins, le protocole Internet définit le format des paquets et fournit un système d'adressage. [12]

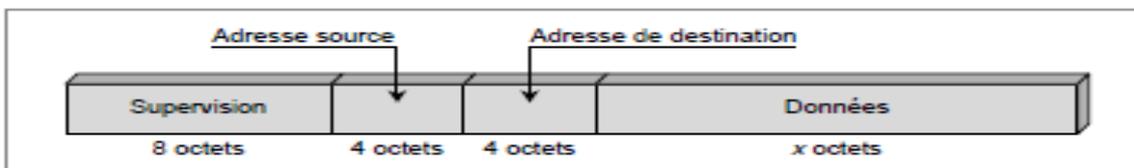


Figure 2.4 Format du Paquet IP

2.4 Modes de transmission :

2.4.1 Mode Unicast :

La monodiffusion est utilisée dans les communications normales d'hôte à hôte tant entre client et serveur que dans un réseau peer-to-peer. Les paquets de type monodiffusion utilisent l'adresse du périphérique de destination comme adresse de destination et peuvent être acheminés sur un inter réseau.

2.4.2 Mode Broadcast :

Le trafic de diffusion est utilisé pour envoyer des paquets à tous les hôtes du réseau grâce à l'adresse de diffusion du réseau. En diffusion, le paquet contient une adresse IPv4 de destination avec uniquement des "1" dans la partie hôte. Cela signifie que tous les hôtes se trouvant sur ce réseau local (domaine de diffusion) recevront le paquet et le regarderont.

2.4.3 Mode Multicast :

La transmission multidiffusion réduit le volume du trafic en permettant à un hôte d'envoyer un seul paquet à un groupe d'hôtes désigné inscrits à un groupe de multidiffusion.

IPv4 a réservé les adresses 224.0.0.0 à 239.255.255.255 comme plage de multidiffusion. Les adresses de multidiffusion IPv4 du bloc 224.0.0.0 à 224.0.0.255 sont réservées à la multidiffusion sur le réseau local uniquement.

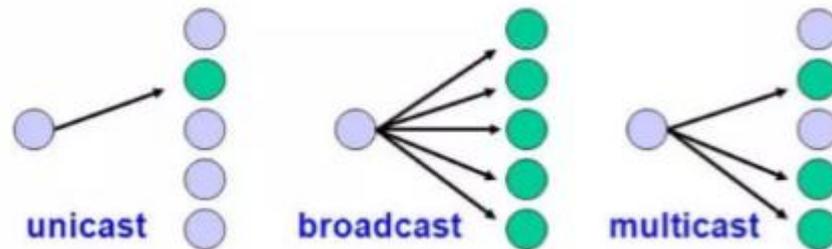


Figure 2.5 Les Modes de transmission

2.5 Les réseaux mobiles ad hoc :

Les réseaux ad hoc auxquels nous nous sommes intéressés sont ceux décrits et étudiés par le groupe de travail MANET (Mobil Ad hoc Network) de l'IETF (Internet Engineering Task Force) en France 2007. Une définition de ces réseaux est donnée formellement dans RFC 2501 : «Un réseau ad hoc comprend des plates-formes mobiles (par exemple, un routeur interconnectant différents hôtes et équipements sans fil) appelées nœuds qui sont libres de se déplacer sans contrainte. Un réseau ad hoc est donc un système autonome de nœuds mobiles. Ce système peut fonctionner d'une manière isolée ou s'interfacer à des réseaux fixes au travers de passerelles». [13]

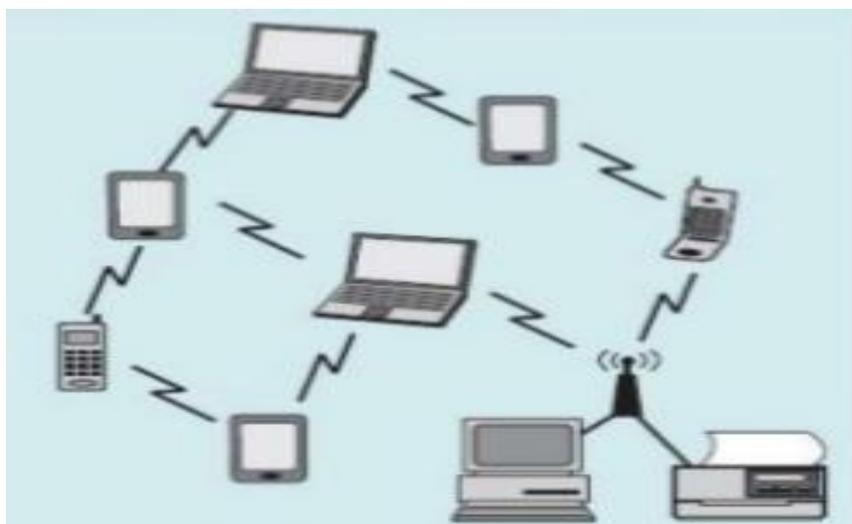


Figure 2.6 Exemple d'une topologie Ad-hoc

2.5.1 Les caractéristiques des réseaux ad-hoc

Parmi les principales caractéristiques des réseaux ad-hoc, nous exposons :

2.5.1.1 Une topologie dynamique :

Les unités mobiles du réseau, se déplacent d'une façon libre et arbitraire. Par conséquent la topologie du réseau peut changer, à des instants imprévisibles, d'une manière rapide et aléatoire. Les liens de la topologie peuvent être unis ou bidirectionnels. [13]

2.5.1.2 Une bande passante limitée :

Une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un hôte soit modeste. [13]

2.6 Hostilités du canal de transmission :

Parmi les critères qui génèrent une hostilité sur le canal de transmission, on trouve :

2.6.1 Le délai :

C'est le temps écoulé entre l'envoi d'un paquet par l'émetteur et sa réception par le destinataire, caractérisé par :

- Délai de traitement
- Délai de mise dans la file d'attente
- Délai de transmission
- Délai de propagation

2.6.2 La latence :

La latence est le temps nécessaire à un paquet pour passer de la source à destination.

2.6.3 La gigue (jitter) :

La gigue (jitter) est la variation du délai de bout en bout entre les différents paquets.

2.6.3.1 La relation entre la gigue et la latence :

La variation de la latence dans le temps est appelé la gigue.

2.6.4 Perte de paquet :

La perte de paquet ou bien le taux de perte de paquet est le pourcentage de paquets perdus lors de la transmission de données.

2.6.5 Le débit :

C'est le taux de transfert maximal entre un émetteur et récepteur.

2.7 Simulateurs et émulateurs de réseaux :

2.7.1 Définition, différences :

2.7.1.1 Simulateur de réseau :

La simulation de réseau est une technique selon laquelle un logiciel modélise le comportement d'un réseau en calculant l'interaction entre les différentes entités du réseau (routeurs, commutateurs, nœuds, points d'accès, liens, etc.). [14]

2.7.1.2 Emulateur de réseau :

L'émulation réseau est une technique permettant de tester les performances d'applications réelles sur un réseau virtuel. Ceci diffère de la simulation de réseau où des modèles purement mathématiques de trafic, des modèles de réseau, des canaux et des protocoles sont appliqués. L'objectif est d'évaluer les performances, de prévoir l'impact du changement ou d'optimiser de toute autre manière la prise de décision technologique. [15]

2.7.1.3 Différences entre simulateur et émulateur de réseau :

L'émulation diffère de la simulation en ce qu'un émulateur de réseau semble être un réseau. Les systèmes d'extrémité tels que les ordinateurs peuvent être connectés à l'émulateur et se comportent comme s'ils étaient connectés à un réseau. Un émulateur de réseau reflète le réseau qui relie les systèmes d'extrémité, pas les systèmes d'extrémité eux-mêmes.

Les simulateurs de réseau sont généralement des programmes exécutés sur un seul ordinateur, décrivant de manière abstraite le trafic réseau tel qu'un processus d'arrivée de flux et générant des statistiques de performances telles que l'occupation d'une mémoire tampon en fonction du temps. [15]

2.7.2 Quelques simulateurs et émulateurs :

Parmi les simulateurs réseaux, on peut citer :

2.7.2.1 NS2 :

NS2 (Network Simulator 2) Est un simulateur de réseau à événements discrets. Il est populaire dans le milieu de la recherche par son caractère extensible, sa nature de logiciel libre et la disponibilité d'une documentation riche sur internet. Ce simulateur est plutôt utilisé pour la simulation du routage et de protocoles d'émission/réception et surtout pour la

recherche dans les réseaux ad-hoc. En effet, NS supporte plusieurs protocoles réseaux et permet la simulation de réseaux sans fils et câblés aussi. [16]

2.7.2.2 OMNeT++ :

OMNeT++ (Objective Modular Network Testbed in C++) est également un partenariat public-source simulateur de réseau à base de composants avec le soutien de l'interface graphique. Son domaine d'application principal est les réseaux de communication. OMNeT++ a une architecture générique et flexible qui fait son succès dans d'autres domaines comme les systèmes informatiques, les réseaux de files d'attente, des architectures matérielles, ou même des processus d'affaires. [17]

Parmi les émulateurs réseaux :

2.7.2.3 NETEM :

Netem (Network Emulator) offre une fonctionnalité d'émulation réseau pour les tests de protocole simulant les propriétés d'un réseau étendu. Actuellement, il prend en charge les délais, pertes, duplications et réarrangements. S'il est exécuté dans une distribution (Fedora, OpenSuse, Gentoo, Debian, Mandriva, Ubuntu) avec le noyau 2.6, il sera déjà activé par défaut. Netem est contrôlé par la commande 'tc' qui fait partie du package iproute2. [18]

2.7.2.4 GNS3 :

GNS3 (Graphical Network Simulator-3) est un émulateur de logiciel réseau publié pour la première fois en 2008 [1] [2] [3]. Il permet la combinaison de périphériques virtuels et réels, utilisés pour simuler des réseaux complexes. Il utilise le logiciel d'émulation Dynamips pour simuler Cisco IOS. [19]

2.8 Conclusion :

Dans ce chapitre, des rappels succincts sur les modèles OSI et TCP/IP, les protocoles de transport UDP TCP/IP et RTP, les modes de transmission et le réseau mobile Ad-hoc avec ses caractéristiques ont été rappelés.

Nous avons aussi vu les différents types de problèmes qui peuvent survenir dans un support de transmission que ce soit filaire ou sans fil. Enfin, des définitions, la différence entre un simulateur et émulateur de réseau et des exemples de ces derniers ont été présentés.

Chapitre 3 : Simulations, résultats et analyses

3.1 Introduction :

Dans ce chapitre, l'étude de l'impact de la qualité du canal de transmission sur les vidéos transmises est présentée. Pour mener les simulations, un réseau sans fil, AD-hoc entre deux ordinateurs a été configuré. Le système d'exploitation utilisé est Linux UBUNTU 18.04LTS. L'émulateur de réseaux "NETEM" a été utilisé pour générer des pertes de paquets aléatoires au niveau du bitstream transmis avec un protocole de transmission RTP. Enfin, le logiciel FFMPEG sert à décoder la vidéo reçue.

Les bitstreams transmis sont ceux de différentes séquences vidéos de différentes résolutions, frame rate et durées. Ils ont été générés par la plateforme HM implémentant le dernier standard de compression vidéo, HEVC/H.265. Différents taux de compression et différentes configurations ont été testés.

Pour étudier l'impact de PLR sur les vidéos transmises, nous estimons la taille des pertes et nous utiliserons la métrique PSNR pour évaluer la détérioration de la qualité entre le fichier transmis et reçu.

3.2 Logiciel et matériel utilisés :

3.2.1 Matériel :

Deux ordinateurs ont été utilisés pour réaliser un réseau ad-hoc sans fils. L'un représente le serveur alors que le deuxième le client. Leurs caractéristiques sont présentées dans le tableau suivant :

Tableau 3.1 Caractéristiques du matériel utilisé

Caractéristiques	Serveur	Client
Type d'ordinateur	Portable	fixe
Processeur	Intel I5 @1.80 GHz	Intel I3 @3.60 GHz
Système d'exploitation (OS)	Linux (UBUNTU 18.04 LTS)	Linux (UBUNTU 18.04 LTS)
Type de système	64 bits	64 bits
RAM	8,00 Go	8,00 Go
DDR	1000 Go (1 To)	500 Go
Carte réseau	WIFI 2.4 GHz	WIFI 2.4 GHz

3.2.2 Logiciels :

- Le système d'exploitation UBUNTU 18.04 LTS
- FFMPEG sur le serveur et client pour la transmission et la réception
- NETEM (Network Simulator) soit au niveau de serveur ou bien au niveau de client, par ce que si nous activons NETEM au niveau client et server on aura une double perte (par exemple, si on applique une perte de 1% au niveau serveur et client on aura une perte totale de 2%).

Dans les sections suivantes, nous résumons la procédure d'installation des logiciels utilisés ainsi que les principales commandes.

3.2.2.1 Installation de FFMPEG :

Pour installer FFMPEG sous UBUNTU 18.04 LTS, tapez :

Sudo apt install ffmpeg (La version installée est FFMPEG 3.3.4)

Si nous voulons installer la dernière version 4.1.1, nous devons télécharger le paquet FFMPEG à partir du ce lien :

https://doc.ubuntu-fr.org/ffmpeg#commencer_sur_une_bonne_base

Pour voir la version installée sur votre noyau, tapez : **ffmpeg -version** tel qu'il est montré dans l'exemple suivant :

```
senoussi@senoussi:~$ ffmpeg -version
ffmpeg version 4.1.1-0york1-18.04 Copyright (c) 2000-2019 the FFmpeg developers
built with gcc 7 (Ubuntu 7.3.0-27ubuntu1-18.04)
configuration: --prefix=/usr --extra-version='0york1-18.04' --toolchain=hardened
--libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-gnu --arc
h=amd64 --enable-gpl --disable-stripping --enable-avresample --disable-filter=re
sample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libaom --enabl
e-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio -
```

Figure 3.1 Commande ffmpeg pour connaître la version installée

3.2.2.2 Quelques commandes FFMPEG :

- Décodage : **ffmpeg -i input.hevc result.yuv**
Cette commande permet de convertir (décoder) le Bitstream HEVC et obtenir un fichier YUV.
- Visualiser la vidéo à partir du Bitstream : **ffplay input.hevc**
- Connaître la version ffmpeg : **ffmpeg -version**
- Calculer le PSNR : **ffmpeg -s resolution -r fps-i video1.yuv -s resolution -r fps-i video2.yuv -lavfi psnr -f null -**

- Calculer le SSIM : **ffmpeg -s resolution -r fps-i video1.yuv -s resolution -r fps-i videop2.yuv -lavfi ssim -f null -**
- Transmission et réception : cette partie nécessite beaucoup plus d'explications et est détaillée dans les sections suivantes :

3.2.2.3 Installation de NETEM :

Le logiciel NETEM (Network Emulator) ne peut être installé que sur le système d'exploitation linux. Dans notre cas, l'émulateur réseau NETEM est installé sur UBUNTU 18.04 LTS et est intégré dans le noyau UBUNTU sans besoin de l'installer. Dans le cas où NETEM n'est pas installé dans le noyau, il faudrait installer le paquet iproute2 qui contient l'émulateur de réseau NETEM. Pour cela, il faut exécuter la commande cmd suivante :

Sudo apt install iproute2

La formule générale :

SYNOPSIS : sudo tc qdisc [...] dev [DEVICE] root netem [OPTIONS]

La commande pour créer une règle : **sudo tc qdisc add**

Pour changer une règle : **sudo tc qdisc change**

Pour remplacer une règle : **sudo tc qdisc replace**

Pour supprimer une règle : **sudo tc qdisc delete**

Ensuite il faut spécifier l'interface réseau : dev **eth0**, pour connaître le nom de l'interface (active), tapez : **ifconfig**

Les options sont: [LIMIT] [DELAY] [LOSS] [CORRUPT] [DUPLICATION]

[REORDERING] [RATE] [SLOT]

3.2.2.4 Quelques commandes NETEM :

- Création de retard (delay) : **sudo tc qdisc add dev eth0 root netem delay 200 ms**
Création d'un retard de transmission à l'interface eth0 égal à 200ms.
- Introduction de pertes de paquet (loss) : **sudo tc qdisc add dev eth0 root netem loss 10%**
Cette commande NETEM permet de générer 10% de perte dans les paquets émis.

- La duplication (duplicate) : **sudo tc qdisc add dev eth0 root netem duplicate 1%**
La duplication de paquets est spécifiée de la même manière que la perte de paquets.
- La corruption (corruption) : **sudo tc qdisc add dev eth2 root netem corrupt 5%**
Un bruit aléatoire peut être émulé comme une option de corruption. Ceci introduit une erreur de bit unique à une position aléatoire dans le paquet.
- La gigue (jitter) : **sudo tc qdisc add dev eth0 root netem delay 100ms 10ms**
Les réseaux étendus réels (WAN) présentent une variabilité. Il est donc possible d'ajouter une variation aléatoire (égale à 10 ms seconde dans cet exemple).
- Réordonner (re-ordering) : **sudo tc qdisc change dev eth2 root netem gap 5 delay 10ms**
Cette commande permet de transmettre immédiatement le 5^{ème}, le 10^{ème}, le 15^{ème}... etc. tandis que les autres paquets seront retardés de 10 ms.

Dans ce projet, nous nous sommes intéressés à l'étude des pertes de paquets considérés comme le problème le plus important dans les canaux de transmission.

3.3 Simulations :

3.3.1 Configuration du réseau :

Pour la transmission des vidéos nous avons configuré un réseau sans fil ad-hoc entre 2 ordinateurs.

Pour activer un point d'accès sans fils sur UBUNTU 18.04 nous devons suivre les étapes suivantes dans l'un des PC :

- Aller dans Paramètres
- Puis WI-FI
- Et puis Activer point d'Access

Puis l'ordinateur démarre un réseau sans fils avec son propre nom et mot de passe alors que le deuxième PC va être simplement connecté au réseau AD-hoc.

Le réseau AD-hoc utilise le protocole de gestion DHCP qui attribue dynamiquement une adresse IP au client/serveur et d'autres paramètres de configuration réseau comme le masque et la passerelle par défaut.

La figure suivante schématise le réseau sans fil créé et montre ses paramètres tels que les adresses IP et les ports utilisés du côté du serveur et du client.



Figure 3.2 Paramètres du réseau sans fil utilisé

3.3.2 Le protocole utilisé :

Pour la transmission de la vidéo avec FFMPEG, nous utilisons le protocole de transport en temps réel RTP qui permet de diffuser de la vidéo sur le réseau IP.

La méthode consiste à préciser l'adresse IP de destination et le port choisi.

Exemple :

`rtp://@10.42.0.1:1234` => l'adresse IP de destination est 10.42.0.1 et le port est 1234

3.3.3 Les vidéos utilisées et leurs caractéristiques :

Dans les tests menés, nous avons choisi trois vidéos différentes en termes de résolutions, fréquences, tailles et durées (nombre de frames). Ces trois vidéos ont été compressées selon les trois configurations : ALL INTRA (AI), LOW DELAY P (LDP) et RANDOM ACCESS (RA).

Tableau 3.2 Caractéristiques des vidéos utilisées

	Résolution	Fps	N _F	Taille Mo	class	extension
BQsquare	416*240	@60	600	90,00	D	YUV
Kimono	1920*1080	@24	240	746,5	B	YUV
RaceHorses	832*480	@30	300	179,7	C	YUV

Fps : frame par seconde N_F : nombre de frame

3.3.4 Procédure de simulation :

Pour la transmission sans fil de la vidéo, nous allons transmettre le bitstream généré par le codeur du plus récent codec, le HEVC/H.265. La plateforme HM a été utilisée dans ce but. Pour chaque vidéo et chaque configuration, des bitstreams sont générés pour différents taux de compression. Ces taux sont déterminés par le paramètre de quantification (QP) choisi égal à 22, 27, 32 et 37 selon les recommandations. [20]

Le FFMPEG est utilisé pour la transmission alors que NETEM est l'outil de contrôle de trafic qui va permettre de générer différentes pertes de paquet. Les simulations ont été menées avec des taux de pertes (PLR= Paquet Loss Rate) de 1%, 3% et 5%. Au-delà de 5%, la qualité de la vidéo devient fortement dégradée. Afin de comparer les effets des pertes de paquets sur la qualité des vidéos compressées, nous avons aussi testé les mêmes pertes sur les vidéos non compressées. Pour cela, les bitstreams ont été générés en procédant à une compression avec QP=0.

Le serveur contient le Bitstream « .hevc » des vidéos à transmettre ainsi qu'un fichier ayant l'extension .SDP qui contient les informations du destinataire et de l'expéditeur (exemple gold.sdp).

Le PC client doit aussi contenir le même fichier .SDP pour une transmission sans erreur (gold.sdp).

```
SDP:
v=0
o=- 0 0 IN IP4 10.42.0.43
s=No Name
c=IN IP4 10.42.0.1
t=0 0
a=tool:libavformat 58.20.100
m=video 1234 RTP/AVP 96
a=rtpmap:96 H265/90000
a=fmtp:96 sprop-vps=QAEMAF//AWAAAAAMAAAMAAAMAAAPAK; sprop-sps=QgEBAWAAAAAMAAAMAAAMAAAKANCA8f5fkkbZ7Z; sprop-pps=RAHBkJWBEg==
```



Figure 3.3 Fichier SDP

3.3.4.1 Test d'une transmission fiable PLR=0% :

Nous prenons comme exemple, la transmission du Bitsream de la vidéo BQsquare compressée en mode ALL INTRA avec QP=22 et PLR=0%.

Les commandes de FFMPEG sont les suivants :

- Au niveau client pour la réception :

```

senoussi@ubuntu:~/Bureau/client ffmpeg/all intra/bq$ ffmpeg -protocol_whitelist
"file,rtp,udp" -i gold.sdp -c copy -strict -2 BQsquare22.hevc
ffmpeg version 4.1.3-0york1~18.04 Copyright (c) 2000-2019 the FFmpeg developers
built with gcc 7 (Ubuntu 7.3.0-27ubuntu1~18.04)
configuration: --prefix=/usr --extra-version='0york1~18.04' --toolchain=harden
ed --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-gnu --a
rch=amd64 --enable-gpl --disable-stripping --enable-avresample --disable-filter=
resample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libaom --ena
ble-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio
--enable-libcodecs2 --enable-libflite --enable-libfontconfig --enable-libfreetyp
e --enable-libfribidi --enable-libgme --enable-libgsm --enable-libjack --enable-
libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-
libopus --enable-libpulse --enable-libsvg --enable-librubberband --enable-libsh
ine --enable-libsndio --enable-libsoxr --enable-libspeex --enable-libssh --enab
le-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-
libvpx --enable-libwavpack --enable-libwebp --enable-libx265 --enable-libxml2 --
enable-libxvid --enable-libzmq --enable-libzvb --enable-lv2 --enable-omx --ena
ble-opengl --enable-opengl --enable-sdl2 --enable-nonfree --enable-libfdk-aac --
enable-libdc1394 --enable-libdrm --enable-libiec61883 --enable-chromaprint --ena
ble-frei0r --enable-libx264 --enable-shared
libavutil      56. 22.100 / 56. 22.100
libavcodec     58. 35.100 / 58. 35.100
libavformat    58. 20.100 / 58. 20.100
libavdevice    58.  5.100 / 58.  5.100

```

Figure 3.4 Commande ffmpeg pour la réception du fichier

C.-à-d. le client est prêt à recevoir (en écoute), le protocole utilisé est le RTP, le fichier est gold.sdp et le nom de fichier reçu est (BQsquare22.hevc)

- Au niveau serveur pour la transmission :

```

senoussi@ubuntu:~/Bureau/SERVER/ALL INTRA I I I/BQSquare_416x240_60$ ffmpeg -re
-i BQSquare_416x240_60qp22.hevc -c copy -f rtp rtp://10.42.0.1:1234 > gold.sdp

```

Figure 3.5 Commande ffmpeg pour la transmission

Cette commande indique que le Bitstream sera transmis à @IP destination (10.42.0.1) en utilisant le port (1234) en utilisant le protocole de transport RTP et le fichier gold.sdp

- Une fois la transmission terminée :

Au niveau serveur :

```

frame= 355 fps= 25 q=-1.0 size= 9895kB time=00:00:14.16 bitrate=5724.4kbits/
frame= 372 fps= 25 q=-1.0 size= 10353kB time=00:00:14.84 bitrate=5715.1kbits/
frame= 385 fps= 25 q=-1.0 size= 10698kB time=00:00:15.36 bitrate=5705.3kbits/
frame= 398 fps= 25 q=-1.0 size= 11038kB time=00:00:15.88 bitrate=5693.9kbits/
frame= 410 fps= 25 q=-1.0 size= 11348kB time=00:00:16.36 bitrate=5682.1kbits/
frame= 423 fps= 25 q=-1.0 size= 11677kB time=00:00:16.88 bitrate=5667.1kbits/
frame= 435 fps= 25 q=-1.0 size= 11981kB time=00:00:17.36 bitrate=5653.9kbits/
frame= 448 fps= 25 q=-1.0 size= 12313kB time=00:00:17.88 bitrate=5641.5kbits/
frame= 460 fps= 25 q=-1.0 size= 12621kB time=00:00:18.36 bitrate=5631.2kbits/
frame= 473 fps= 25 q=-1.0 size= 12955kB time=00:00:18.88 bitrate=5621.3kbits/
frame= 485 fps= 25 q=-1.0 size= 13262kB time=00:00:19.36 bitrate=5611.7kbits/
frame= 498 fps= 25 q=-1.0 size= 13590kB time=00:00:19.88 bitrate=5600.0kbits/
frame= 510 fps= 25 q=-1.0 size= 13890kB time=00:00:20.36 bitrate=5588.6kbits/
frame= 523 fps= 25 q=-1.0 size= 14208kB time=00:00:20.88 bitrate=5574.5kbits/
frame= 536 fps= 25 q=-1.0 size= 14523kB time=00:00:21.40 bitrate=5559.5kbits/
frame= 548 fps= 25 q=-1.0 size= 14812kB time=00:00:21.88 bitrate=5545.6kbits/
frame= 561 fps= 25 q=-1.0 size= 15125kB time=00:00:22.40 bitrate=5531.3kbits/
frame= 574 fps= 25 q=-1.0 size= 15439kB time=00:00:22.92 bitrate=5518.2kbits/
frame= 586 fps= 25 q=-1.0 size= 15728kB time=00:00:23.40 bitrate=5506.1kbits/
frame= 599 fps= 25 q=-1.0 size= 16040kB time=00:00:23.92 bitrate=5493.2kbits/
frame= 600 fps= 25 q=-1.0 Lsize= 16064kB time=00:00:23.96 bitrate=5492.2kbits
/s speed= 1x
video:15899kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing
overhead: 1.035393%

```

Figure 3.6 Indication ffmpeg sur la fin de la transmission

Au niveau client :

```

frame= 491 fps= 26 q=-1.0 size= 13056kB time=00:00:19.56 bitrate=5468.0kbits/
frame= 504 fps= 26 q=-1.0 size= 13568kB time=00:00:20.08 bitrate=5535.3kbits/
frame= 517 fps= 26 q=-1.0 size= 13824kB time=00:00:20.60 bitrate=5497.4kbits/
frame= 530 fps= 26 q=-1.0 size= 14080kB time=00:00:21.12 bitrate=5461.3kbits/
frame= 543 fps= 26 q=-1.0 size= 14336kB time=00:00:21.64 bitrate=5427.0kbits/
frame= 556 fps= 26 q=-1.0 size= 14848kB time=00:00:22.16 bitrate=5488.9kbits/
frame= 569 fps= 26 q=-1.0 size= 15104kB time=00:00:22.68 bitrate=5455.6kbits/
frame= 582 fps= 26 q=-1.0 size= 15360kB time=00:00:23.20 bitrate=5423.7kbits/
frame= 595 fps= 26 q=-1.0 size= 15616kB time=00:00:23.72 bitrate=5393.2kbits/
gold.sdp: Connection timed out
frame= 600 fps= 18 q=-1.0 size= 15872kB time=00:00:23.92 bitrate=5435.8kbits/
frame= 600 fps= 14 q=-1.0 Lsize= 15900kB time=00:00:23.92 bitrate=5445.2kbits/
/s speed=0.555x
video:15900kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing
overhead: 0.000000%
senoussi@ubuntu:~/Bureau/client ffmpeg/all intra/bq$

```

Figure 3.7 Indication ffmpeg sur la réception

A la fin, on trouve le bitstream au niveau d'un dossier de client. Il est alors converti (décodé) en format YUV en utilisant la commande ffmpeg suivante :

ffmpeg -i BQsquare22.hevc BQsquare22.yuv



Figure 3.8 Les fichiers au niveau client

Les résultats sont présentés dans le tableau suivant :

Tableau 3.3 Résultats du test de la transmission

	Taille fichier HEVC transmis [Mo]	Taille fichier HEVC reçu [Mo]	Taille fichier YUV transmis [Mo]	Taille fichier YUV reçu [Mo]
BQsquare416*240@60, AI, QP22, PLR=0%	16,3	16,3	90,00	90,00

3.3.4.2 générer une perte de paquet :

Voici un exemple, comment générer une perte de paquet égale 1%. En utilisant le NETEM, tapez :

```
senoussi@ubuntu:~$ sudo tc qdisc add dev wlp8s0 root netem loss 1%
[sudo] Mot de passe de senoussi :
senoussi@ubuntu:~$
```

Figure 3.9 Commande NETEM pour un PLR 1%

Et le résultat sera au niveau client comme ci-dessous :

```
[sdp @ 0x563710c81a80] RTP: missed 1 packets
[sdp @ 0x563710c81a80] max delay reached. need to consume packet
[sdp @ 0x563710c81a80] RTP: missed 1 packets
[sdp @ 0x563710c81a80] max delay reached. need to consume packet
[sdp @ 0x563710c81a80] RTP: missed 1 packets
frame= 568 fps= 26 q=-1.0 size= 15104kB time=00:00:23.00 bitrate=5379.6kbits/
[sdp @ 0x563710c81a80] max delay reached. need to consume packet
[sdp @ 0x563710c81a80] RTP: missed 1 packets
[sdp @ 0x563710c81a80] max delay reached. need to consume packet
[sdp @ 0x563710c81a80] RTP: missed 1 packets
[sdp @ 0x563710c81a80] max delay reached. need to consume packet
[sdp @ 0x563710c81a80] RTP: missed 1 packets
[sdp @ 0x563710c81a80] max delay reached. need to consume packet
[sdp @ 0x563710c81a80] RTP: missed 1 packets
[sdp @ 0x563710c81a80] max delay reached. need to consume packet
[sdp @ 0x563710c81a80] RTP: missed 1 packets
frame= 582 fps= 26 q=-1.0 size= 15360kB time=00:00:23.56 bitrate=5340.8kbits/
gold.sdp: Connection timed out
frame= 591 fps= 18 q=-1.0 size= 15616kB time=00:00:23.92 bitrate=5348.1kbits/
frame= 591 fps= 14 q=-1.0 Lsize= 15743kB time=00:00:23.92 bitrate=5391.7kbits
/s speed=0.555x
video:15743kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing
overhead: 0.000000%
senoussi@ubuntu:~/Bureau/client ffmpeg/all intra/bq$
```

Figure 3.10 Pertes des paquets au niveau de la réception

3.4 Résultats :

Les tableaux suivants résument les résultats des simulations de transmission des bitstreams des vidéos. Rappelons que les vidéos utilisées ont des résolutions différentes qu'elles ont été compressées selon différentes configurations (AI, LDP, RA) et différents paramètres de quantification QP. Du côté réception, nous avons évalué les tailles reçues avec différents PLR, les pertes absolues, les pertes relatives, et la qualité à travers la mesure du PSNR.

3.4.1 En mode All Intra :

Tableau 3.4 Résultat de la transmission de la vidéo BQsquare en mode ALL Intra

BQsquare 416*240@60, All Intra	QP 22	QP 27	QP 32	QP 37
Taille Fichier HEVC transmis (MO)	16.3	10.3	6.4	3.8
Taille fichier HEVC reçu (MO) (perte 0%)	16.3	10.3	6.4	3.8
Taille fichier HEVC reçu (MO) (Perte 1%)	16.132	10.222	6.303	3.792
Perte absolue (MO)	0.168	0.078	0.097	0.008
Perte relative (%)	1.03	0.76	1.51	0.21
PSNR (dB)	16.72	18.09	19.61	18.69
Taille fichier HEVC reçu (MO) (Perte 3%)	15.814	9.990	6.184	3.645
Perte absolue (MO)	0.486	0.31	0.216	0.155
Perte relative (%)	2.98	3	3.37	4.08
PSNR (dB)	15.25	14.78	15.62	16.38
Taille fichier HEVC reçu (MO) (Perte 5%)	15.494	9.793	6.076	3.591
Perte absolue (MO)	0.806	0.502	0.324	0.209
Perte relative (%)	4.95	4.87	5.06	5.5
PSNR (dB)	13.26	14.27	14.54	14.87

Tableau 3.5 Résultat de la transmission de la vidéo Kimono en mode ALL Intra

Kimono 1920*1080@24, All Intra	QP 22	QP 27	QP 32	QP 37
Taille Fichier HEVC transmis (MO)	27.8	15.2	8.6	4.8
Taille fichier HEVC reçu (MO) (perte 0%)	27.8	15.2	8.6	4.8
Taille fichier HEVC reçu (MO) (Perte 1%)	27.512	15.044	8.456	4.742
Perte absolue (MO)	0.288	0.156	0.144	0.058
Perte relative (%)	1.03	1.02	1.51	1.67
PSNR (dB)	18.67	21.34	20.88	24.54
Taille fichier HEVC reçu (MO) (Perte 3%)	26.97	14.76	8.286	4.618
Perte absolue (MO)	0.83	0.44	0.314	0.182
Perte relative (%)	2.98	2.89	3.65	3.79
PSNR (dB)	16.13	15.84	18.68	17.97
Taille fichier HEVC reçu (MO) (Perte 5%)	26.341	14.395	8.126	4.533
Perte absolue (MO)	1.459	0.805	0.474	0.267
Perte relative (%)	5.24	5.29	5.51	5.56
PSNR (dB)	12.83	14.20	15.29	16.38

Tableau 3.6 Résultat de la transmission de la vidéo RaceHorses en mode ALL Intra

RaceHorses 832*480@30, All Intra	QP 22	QP 27	QP 32	QP 37
Taille Fichier HEVC transmis (MO)	18.8	11.3	6.4	3.2
Taille fichier HEVC reçu (MO) (perte 0%)	18.8	11.3	6.4	3.2
Taille fichier HEVC reçu (MO) (Perte 1%)	18.546	11.162	6.331	3.189
Perte absolue (MO)	0.254	0.138	0.069	0.011
Perte relative (%)	1.35	1.22	1.07	0.34
PSNR (dB)	17.29	20.45	20.89	23.11
Taille fichier HEVC reçu (MO) (Perte 3%)	18.528	10.924	6.186	3.146
Perte absolue (MO)	0.272	0.376	0.214	0.054
Perte relative (%)	1.44	3.32	3.34	1.68
PSNR (dB)	15.88	14.91	16.79	16.98
Taille fichier HEVC reçu (MO) (Perte 5%)	17.854	10.634	6.016	3.066
Perte absolue (MO)	0.946	0.666	0.384	0.134
Perte relative (%)	5.03	5.89	6	4.18
PSNR (dB)	13.42	14.24	14.05	16.58

3.4.2 En mode Low Delay P :

Tableau 3.7 Résultat de la transmission de la vidéo BQsquare en mode Low Delay P

BQsquare 416*240@60, Low Delay P	QP 22	QP 27	QP 32	QP 37
Taille Fichier HEVC transmis (MO)	3.5	1.2	458.6 Ko	200.3 Ko
Taille fichier HEVC reçu (MO) (perte 0%)	3.5	1.2	458.6 Ko	200.3 Ko
Taille fichier HEVC reçu (MO) (Perte 1%)	3.449	1.159	453.7	198.809
Perte absolue (MO)	0.051	0.041	4.9	1.421
Perte relative (%)	1.45	3.41	1.06	0.7
PSNR (dB)	12.79	13.64	14.13	13.85
Taille fichier HEVC reçu (MO) (Perte 3%)	3.373	1.132	447.074	194.161
Perte absolue (MO)	0.127	0.068	11.526	6.139
Perte relative (%)	3.62	5.66	2.51	3.06
PSNR (dB)	14.19	13.1	12.76	14.04
Taille fichier HEVC reçu (MO) (Perte 5%)	3.319	1.118	343.084	192.26
Perte absolue (MO)	0.181	0.082	24.516	8.04
Perte relative (%)	5.17	6.83	5.34	4.01
PSNR (dB)	13.65	12.82	13.82	13.75

Tableau 3.8 Résultat de la transmission de la vidéo Kimono en mode Low Delay P

Kimono 1920*1080@24, Low Delay P	QP 22	QP 27	QP 32	QP 37
Taille Fichier HEVC transmis (MO)	6.9	3.1	1.5	749.4 Ko
Taille fichier HEVC reçu (MO) (perte 0%)	6.9	3.1	1.5	749.4 Ko
Taille fichier HEVC reçu (MO) (Perte 1%)	6.804	3.091	1.495	739.621
Perte absolue (MO)	0.096	0.009	0.005	9.779
Perte relative (%)	1.39	0.29	0.33	1.3
PSNR (dB)	16.63	13.67	13.61	18.65
Taille fichier HEVC reçu (MO) (Perte 3%)	6.634	3.026	1.462	730.472
Perte absolue (MO)	0.266	0.074	0.038	18.928
Perte relative (%)	3.85	2.38	2.53	2.52
PSNR (dB)	12.03	11.58	12.51	14.76
Taille fichier HEVC reçu (MO) (Perte 5%)	6.484	2.949	1.425	718.242
Perte absolue (MO)	0.416	0.151	0.075	31.158
Perte relative (%)	6.02	4.87	5	4.15
PSNR (dB)	12.45	12.83	12.22	14.05

Tableau 3.9 Résultat de la transmission de la vidéo RaceHorses en mode Low Delay P

RaceHorses 832*480@30, Low Delay P	QP 22	QP 27	QP 32	QP 37
Taille Fichier HEVC transmis (MO)	7.6	3	1.3	597.7 Ko
Taille fichier HEVC reçu (MO) (perte 0%)	7.6	3	1.3	597.7 Ko
Taille fichier HEVC reçu (MO) (Perte 1%)	7.513	2.934	1.279	593.962
Perte absolue (MO)	0.087	0.066	0.021	3.738
Perte relative (%)	1.14	2.2	1.61	0.62
PSNR (dB)	13.805	14.29	14.34	16.04
Taille fichier HEVC reçu (MO) (Perte 3%)	7.375	2.881	1.269	576.27
Perte absolue (MO)	0.225	0.119	0.031	21.43
Perte relative (%)	2.96	3.96	2.38	3.58
PSNR (dB)	14.55	13.78	15.493	15.86
Taille fichier HEVC reçu (MO) (Perte 5%)	7.226	2.808	1.225	575.158
Perte absolue (MO)	0.374	0.192	0.075	22.542
Perte relative (%)	4.92	6.4	5.76	3.77
PSNR (dB)	13.807	13.53	15.494	15.63

3.4.3 En mode Random Access :

Tableau 3.10 Résultat de la transmission de la vidéo BQsquare en mode Random Access

BQsquare 416*240@60, Random Access	QP 22	QP 27	QP 32	QP 37
Taille Fichier HEVC transmis (MO)	2.083	832.8 Ko	392.5 Ko	211.2 Ko
Taille fichier HEVC reçu (MO) (perte 0%)	2.083	832.8 Ko	392.5 Ko	211.2 Ko
Taille fichier HEVC reçu (MO) (Perte 1%)	2.064	818.605	389.666	209.351
Perte absolue (MO)	0.019	14.195	2.834	1.849
Perte relative (%)	0.91	1.7	0.72	0.87
PSNR (dB)	18.83	16.24	18.36	19.01
Taille fichier HEVC reçu (MO) (Perte 3%)	2.019	803.294	378.114	201.616
Perte absolue (MO)	0.064	29.506	5.386	9.584
Perte relative (%)	3.07	3.54	1.37	4.53
PSNR (dB)	13.96	15.57	16.05	14.82
Taille fichier HEVC reçu (MO) (Perte 5%)	1.984	794.130	369.163	199.557
Perte absolue (MO)	0.099	38.67	32.337	11.643
Perte relative (%)	4.75	3.44	8.23	5.51
PSNR (dB)	13.49	14.15	14	15.13

Tableau 3.11 Résultat de la transmission de la vidéo Kimono en mode Random Access

Kimono 1920*1080@24, Random Access	QP 22	QP 27	QP 32	QP 37
Taille Fichier HEVC transmis (MO)	5.6	2.5	1.2	601 Ko
Taille fichier HEVC reçu (MO) (perte 0%)	5.6	2.5	1.2	601 Ko
Taille fichier HEVC reçu (MO) (Perte 1%)	5.540	2.494	1.185	595.256
Perte absolue (MO)	0.06	0.006	0.015	5.744
Perte relative (%)	1.07	0.24	1.25	0.95
PSNR (dB)	19.03	20	20.71	23.33
Taille fichier HEVC reçu (MO) (Perte 3%)	5.385	2.451	1.169	588.2
Perte absolue (MO)	0.215	0.042	0.031	12.8
Perte relative (%)	3.83	1.68	2.58	2.12
PSNR (dB)	14.26	14.47	17.76	24.52
Taille fichier HEVC reçu (MO) (Perte 5%)	5.293	2.402	1.152	571.276
Perte absolue (MO)	0.307	0.098	0.048	29.724
Perte relative (%)	5.48	3.92	4	4.94
PSNR (dB)	13	12.33	17.8	11.48

Tableau 3.12 Résultat de la transmission de la vidéo RaceHorses en mode Random Access

RaceHorses 832*480@30, Random Access	QP 22	QP 27	QP 32	QP 37
Taille Fichier HEVC transmis (MO)	6	2.6	1.2	597.1 Ko
Taille fichier HEVC reçu (MO) (perte 0%)	6	2.6	1.2	597.1 Ko
Taille fichier HEVC reçu (MO) (Perte 1%)	5.936	2.534	1.185	593.125
Perte absolue (MO)	0.064	0.066	0.015	3.975
Perte relative (%)	1.06	2.53	1.25	0.66
PSNR (dB)	16.97	16.93	16.52	17.62
Taille fichier HEVC reçu (MO) (Perte 3%)	5.836	2.476	1.154	577.954
Perte absolue (MO)	0.164	0.125	0.046	19.146
Perte relative (%)	2.73	4.8	3.83	3.2
PSNR (dB)	16.33	14.29	15.64	17.32
Taille fichier HEVC reçu (MO) (Perte 5%)	5.702	2.436	1.138	556.27
Perte absolue (MO)	0.298	0.164	0.062	40.83
Perte relative (%)	4.96	6.3	5.16	6.83
PSNR (dB)	14.29	15.8	13.91	15.18

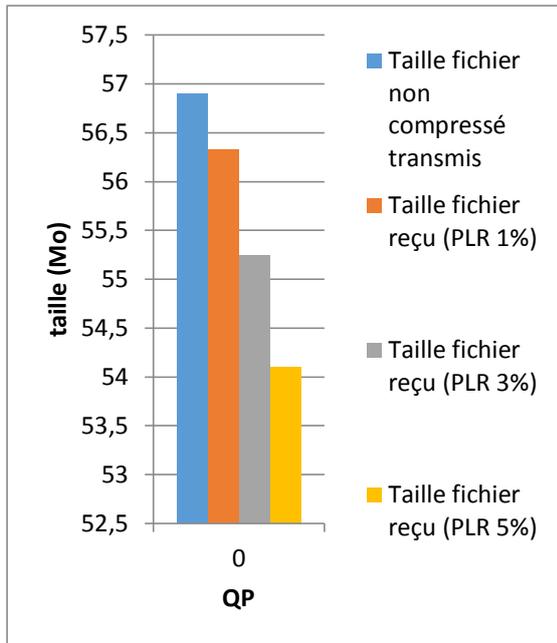
3.5 Analyses et discussions :

3.5.1 Impact du PLR sur la taille du Bitstream :

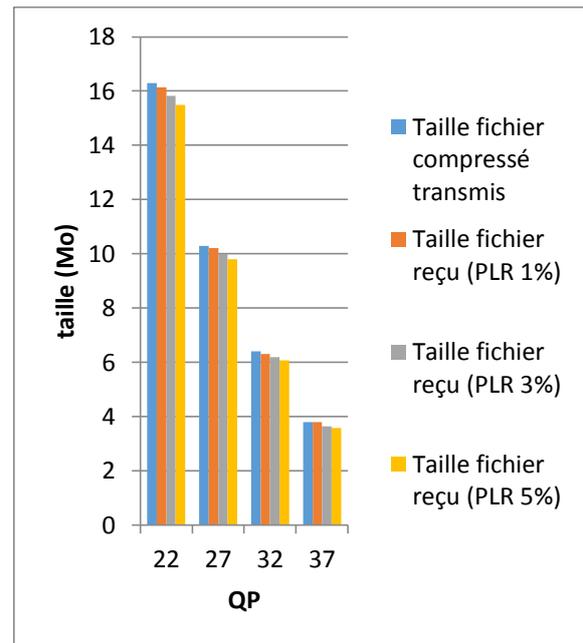
Les figures suivantes représentent les tailles des fichiers transmis et reçus avec différents PLR. On remarque qu'à chaque fois que le PLR augmente (1%, 3%, et 5%), nous observons une diminution (perte) de la taille du fichier vidéo reçu, ce qui laisse prévoir une perte de qualité de la vidéo.

Ce résultat est obtenu quel que soit le taux de compression (QP22, QP27, QP 32, QP37) appliqué, ou bien la configuration utilisée (AI, LD P, RA).

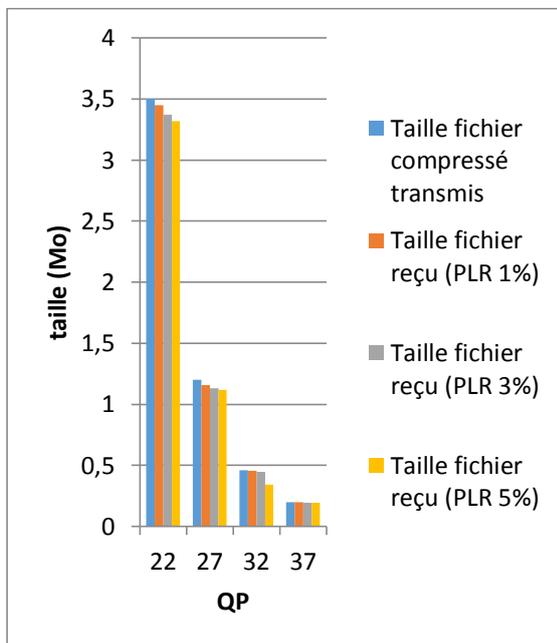
Les pertes sont évidentes dans le cas des fichiers non compressée mais le sont moins lorsque le taux de compression est important (QP élevé). Ceci s'explique tout simplement par le fait que la taille du fichier est de plus en plus petite lorsque QP croit. Mais les pertes restent pratiquement constantes et varient plus ou moins autour des valeurs 1%, 3% et 5%. Il faut, peut-être, rappeler, que les pertes se produisent au niveau des paquets.



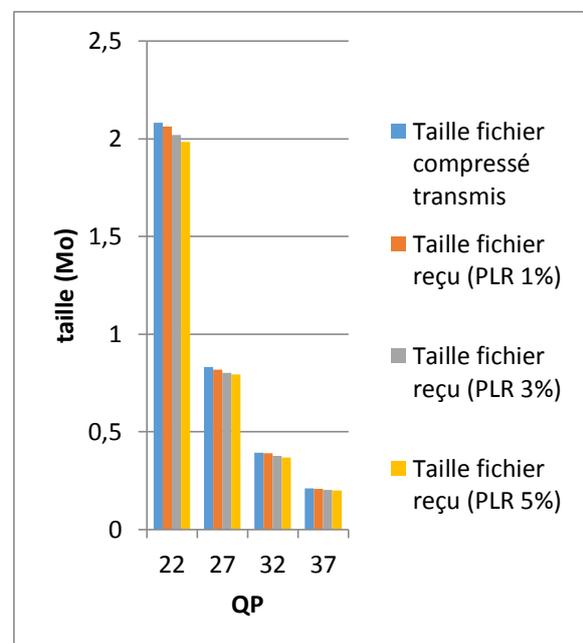
(a)



(b)



(c)

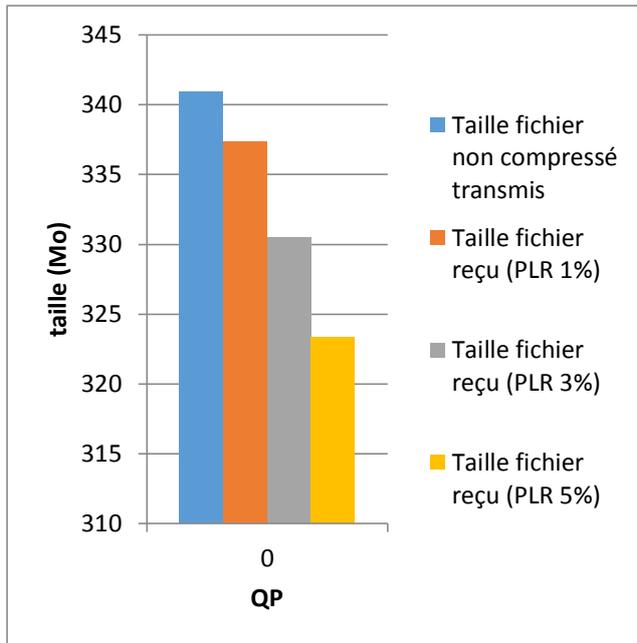


(d)

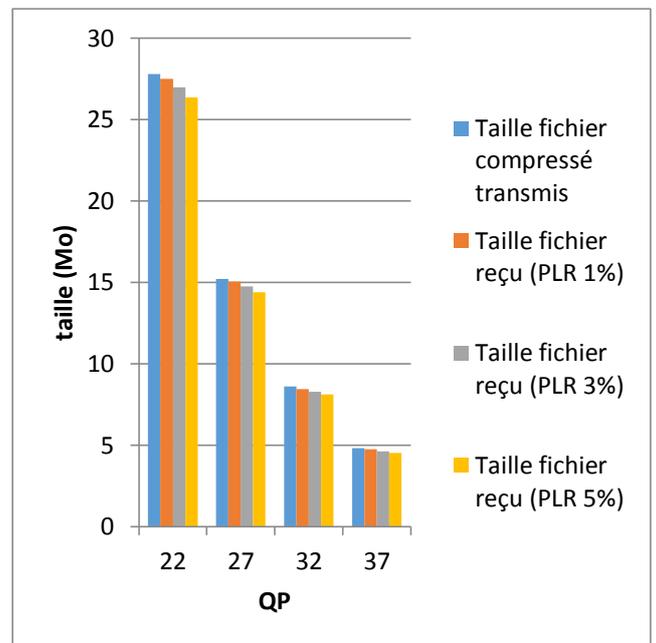
Figure 3.11 Taille des fichiers transmis et reçus avec différents PLR (vidéo BQsquare)

(a) Fichiers non compressés (b) Fichiers compressés (AI)

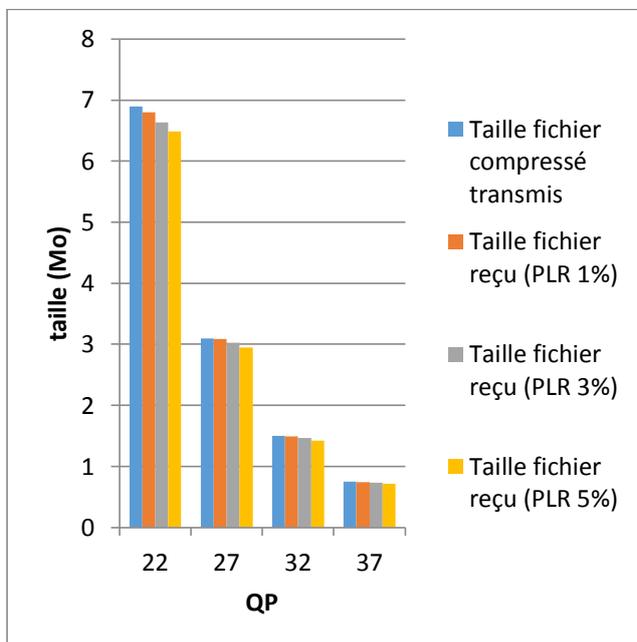
(c) Fichiers compressés (LDP) (d) Fichiers compressés (RA)



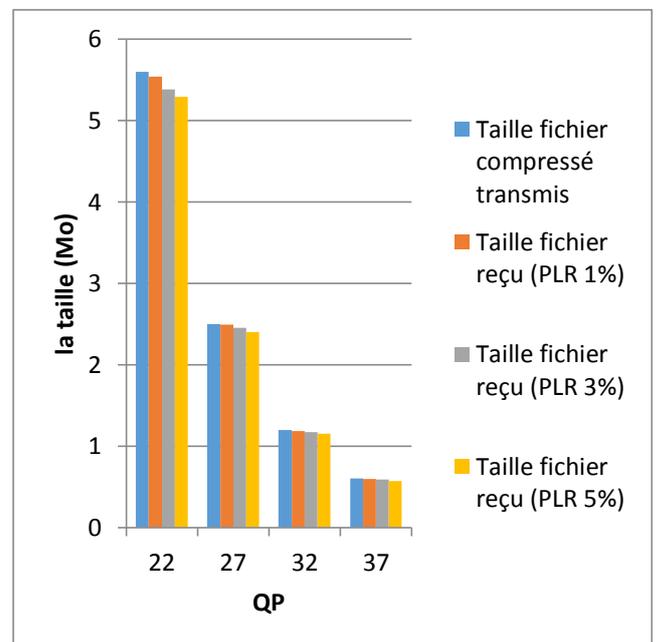
(a)



(b)



(c)

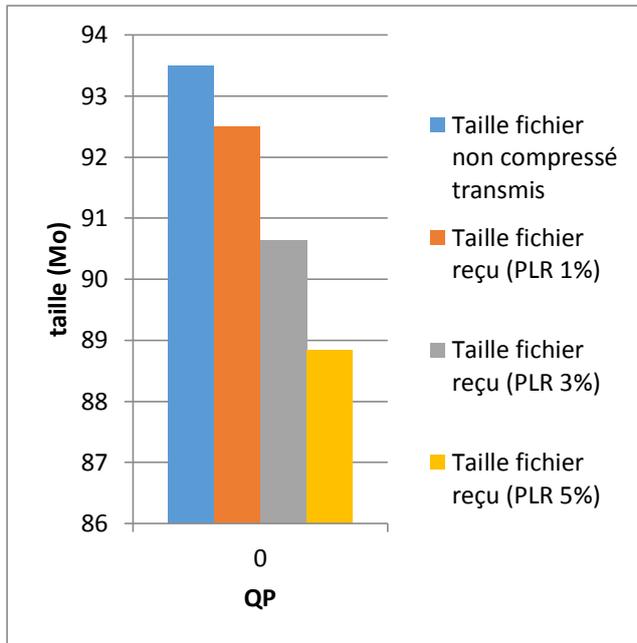


(d)

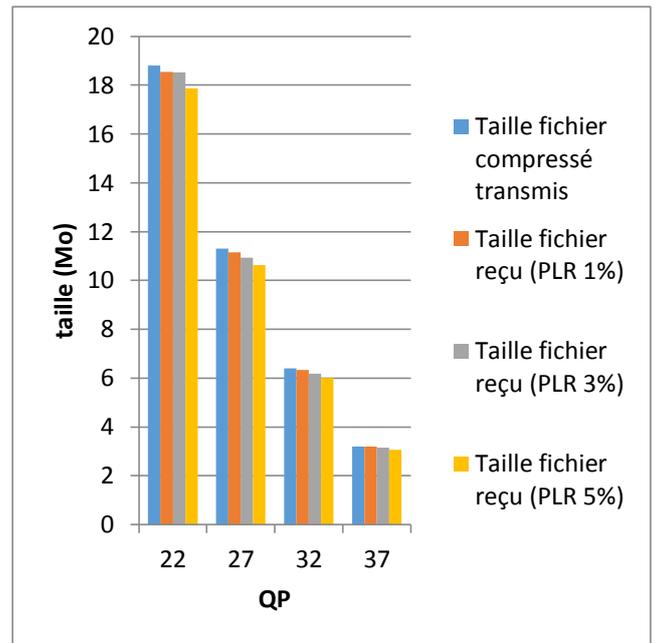
Figure 3.12 Taille des fichiers transmis et reçus avec différents PLR (vidéo Kimono)

(a) Fichiers non compressés (b) Fichiers compressés (AI)

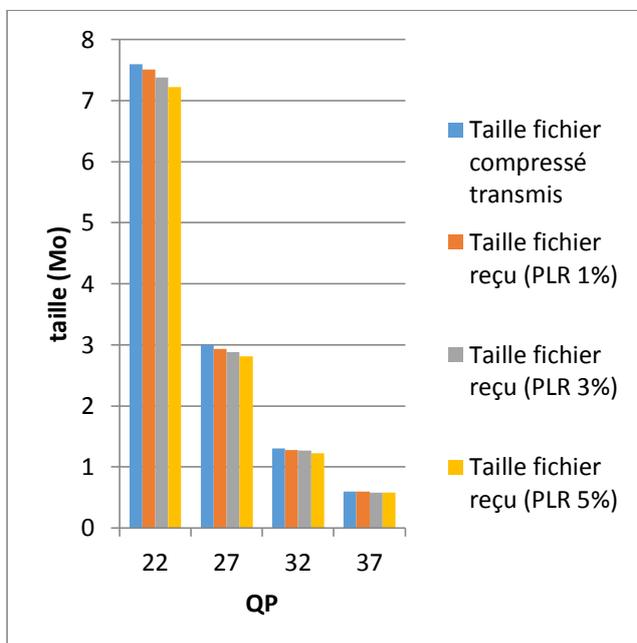
(c) Fichiers compressés (LDP) (d) Fichiers compressés (RA)



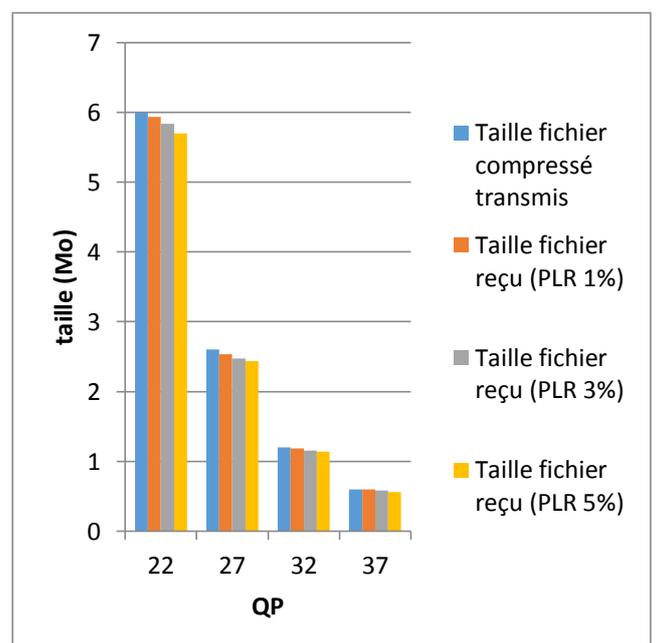
(a)



(b)



(c)



(d)

Figure 3.13 Taille des fichiers transmis et reçus avec différents PLR (vidéo RaceHorses)

(a) Fichiers non compressés (b) Fichiers compressés (AI)

(c) Fichiers compressés (LDP) (d) Fichiers compressés (RA)

3.5.2 Impact du PLR sur la taille perdue (absolue) :

Les figures suivantes représentent la taille des pertes en fonction de QP. On peut faire 2 analyses, une avec le même QP et la 2^{ème} avec les différents QP.

Nous commençons avec l'analyse avec le même QP :

On remarque qu'à chaque fois que le PLR augmente (1%, 3%, et 5%), la taille des pertes augmente et que ceci est valable quelque soit le taux de compression (QP).

2^{ème} analyse avec les différents QP :

On remarque qu'à avec un même PLR et différents QP (22, 27, 32, 37), on aura une diminution de la taille de perte. Ce qui est tout à fait normal et logique. Par exemple, avec QP22, la taille du Bitstream est plus grande que celles relatives aux autres QP (27, 32 et 37). Donc si on applique un PLR de 1% aux Bitstreams de QP22, QP27, QP32 et QP37, on trouve que la taille des pertes de QP22 est plus grande que celle de QP27 qui est à son tour supérieure à la perte avec QP32 et en fin QP37. Les figures suivantes montrent bien que les tailles des pertes absolues sont décroissantes en fonction du QP et ceci pour un même PLR.

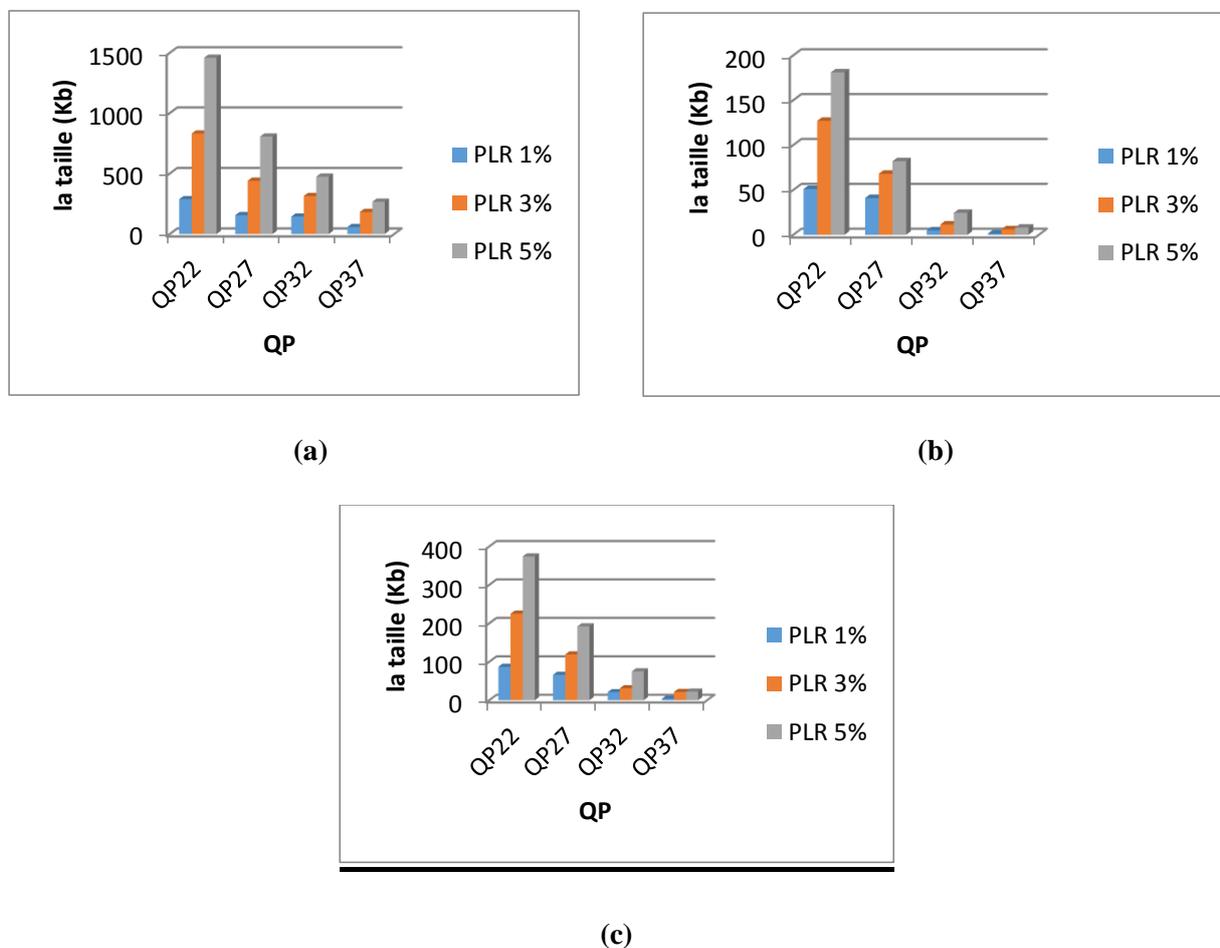


Figure 3.14 Pertes absolues en fonction de QP et PLR

(a) (Kimono, AI), (b) (BQsquare, LDP), (c) (RaceHorses, LDP)

Mais il y a aussi des résultats surprenants, représentés dans les figures ci-dessous :

Si on compare pour un même QP, tous les cas sont logiques quelle que soit la configuration de vidéo ou bien le taux de compression utilisé. Mais pour un même PLR et différents QP, on remarque que les pertes sont de plus en plus faibles.

On prend par exemple le cas de Bitstream BQsquare avec la configuration All Intra :

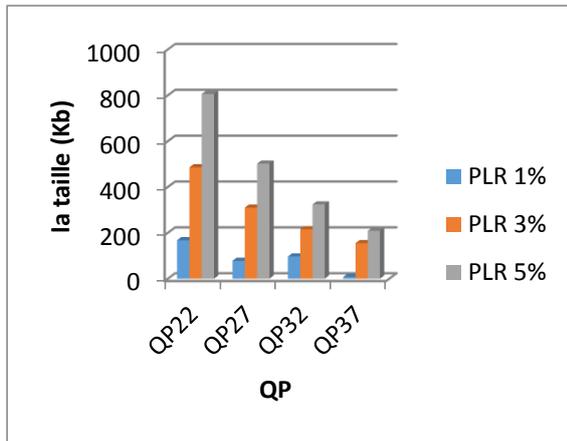
Avec un PLR de 3% et 5%, il est tout à fait normal la taille des pertes va diminuer si le taux de compression est élevé. Mais avec un PLR de 1%, la taille des pertes va diminuer entre QP22 et QP27 (168Kb-78Kb) puis augmenter (97 Kb) pour QP32 puis elle va diminuer normalement (8 Kb) pour QP=37.

Ceci peut s'expliquer par le fait que :

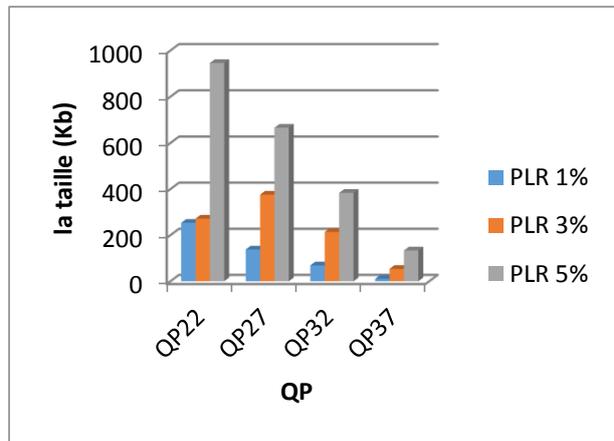
- Relation entre le PLR et la perte en taille du fichier reçu : le PLR généré avec l'outil NETEM est un pourcentage du nombre des paquets transmis. D'après les résultats, les pertes relatives dans les tailles des fichiers des vidéos correspondent plus ou moins au PLR spécifié. On a obtenu une perte variant entre 0.5% et 1.5%.

Par exemple, dans le cas de la vidéo BQsquare et la configuration All Intra, avec un taux de QP=22, la taille du fichier transmis est de 16.3 Mo. Le fichier reçu a perdu 168 Kb de sa taille. Sachant que le PLR utilisé est de 1%, on peut vérifier que cette perte est bien égale à : $16.3\text{Mo} \times (1/100)$ qui égale 163Kb.

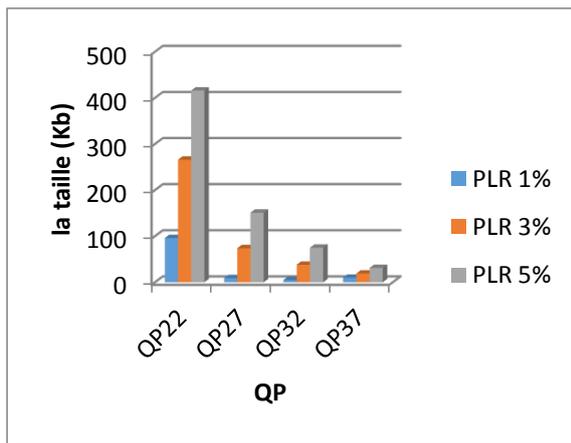
- NETEM utilise une perte de paquets aléatoire. La perte peut se produire au niveau détruit d'un ou plusieurs paquets très importants appartenant à des frames clés et très importants pour la décompression. Par exemple, si des paquets des frames de référence I sont perdus, tous les autres frames seront touchés dans le cas d'une configuration low delay P et tous les frames du GOP seront altérés dans le cas d'une configuration random access. En plus de ça, il faut se rappeler que la taille d'un frame I est plus grande que celles des frames P et B.



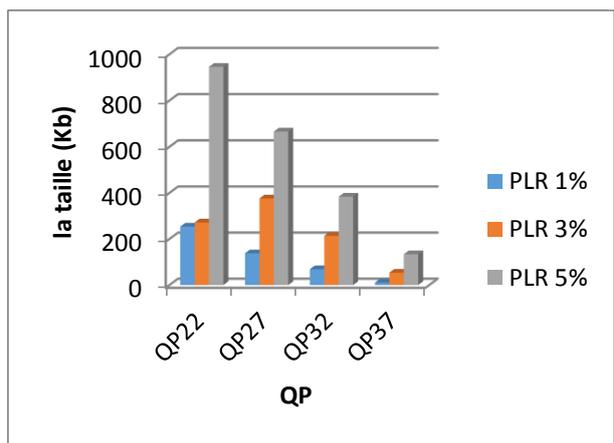
(a)



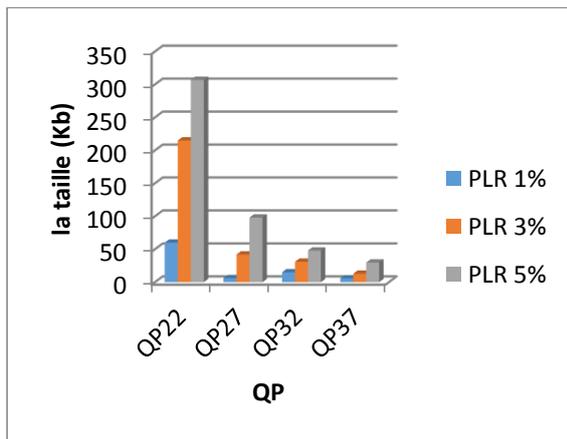
(b)



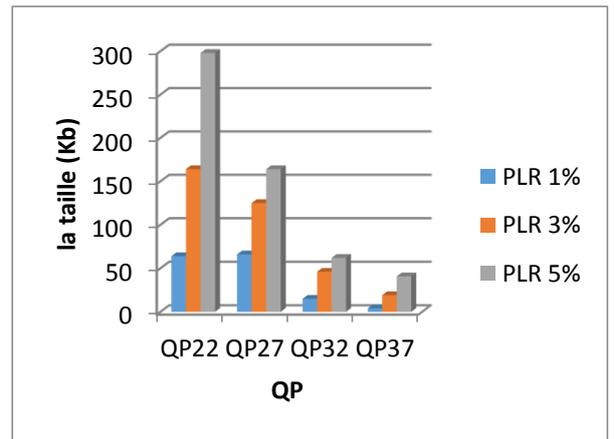
(c)



(d)



(e)



(f)

Figure 3.15 Pertes absolues en fonction de QP et PLR

(a) (BQsquare, AI), (b) (Race Horses, AI), (c) (Kimono, LD P), (d) (BQsquare, RA),
 (e) (Kimono,RA), (f) (Race Horses, RA)

3.5.3 Etude de l'impact du PLR sur la qualité des vidéos :

Les figures suivantes représentent l'effet du PLR sur la qualité des vidéos reconstruites à travers le calcul du PSNR pour différentes configurations et différents taux de compression.

Comme prévu, on remarque que dans le cas sans pertes (PLR= 0%), les PSNR des vidéos diminuent au fur et à mesure que QP augmente. Lorsque le PLR augmente, le PSNR diminue fortement, parce qu'à chaque fois qu'on compresse plus nous avons une diminution plus grande en termes de PSNR c'est-à-dire une dégradation de la qualité de vidéo.

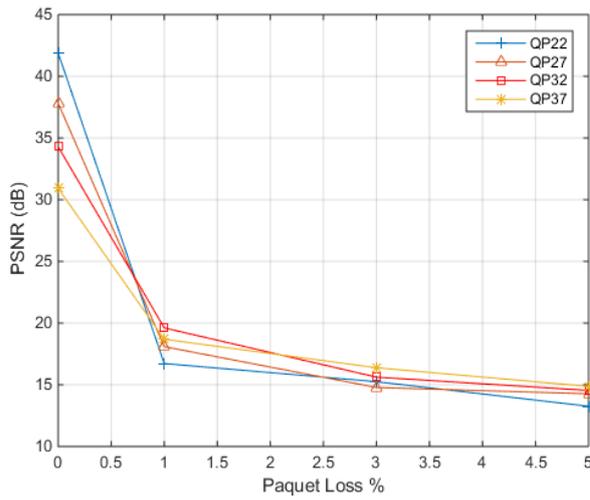
En revanche, dans certains cas, nous avons une dégradation totale bien que le PLR utilisé soit très petit. C'est, par exemple, le cas de la séquence Kimono en mode All Intra avec QP=22. Le PSNR dans le cas sans pertes de paquets (PLR= 0%) est de 43.34dB. Avec un PLR de 1%, le PSNR chute à 18.67dB, produisant une dégradation totale. Le PSNR est de 16.13dB pour un PLR de 3% et de 12.83dB pour un de PLR 5%.

Comme mentionné précédemment, avec une transmission fiable (PLR= 0%), la relation $PSNR(QP22) > PSNR(QP27) > PSNR(QP32) > PSNR(QP37)$ est toujours vérifiée. Mais pour un PLR différent de 0%, les résultats ne correspondent pas toujours à ce qui a été prévu.

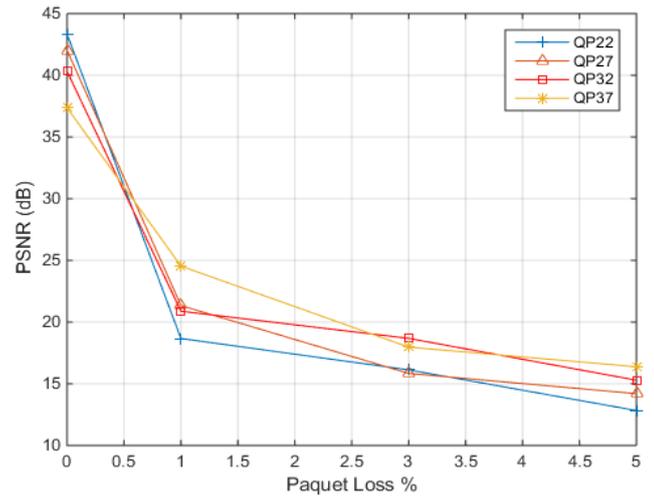
Parmi les raisons possibles :

- La perte aléatoire produite par le NETEM.
- Le type de configuration utilisé :
 1. Le type Low Delay P contient un seul frame I de référence et tout le reste des frames P, donc lorsque la perte est située dans un frame I, tous les frames suivants seront affectés.
 2. Le type Random Access : composé d'une image référence I dans chaque GOP et d'une séquence des frames P/B (intra period). Donc si la perte se produit dans un frame P/B, elle affectera moins la qualité que si elle se situe dans un frame I.
 3. Le type All Intra : tous les frames sont des I et la perte affecte un frame sans conséquences pour le reste des frames. La dégradation de la qualité dépendra du nombre de frames touchés.

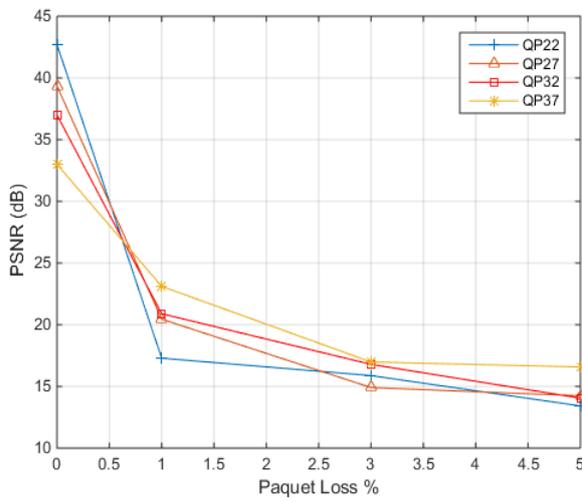
A remarquer qu'avec un PLR 5%, la relation entre les PSNR s'inverse complètement et on a tous jours le $PSNR(QP22) < PSNR(QP27) < PSNR(QP32) < PSNR(QP37)$.



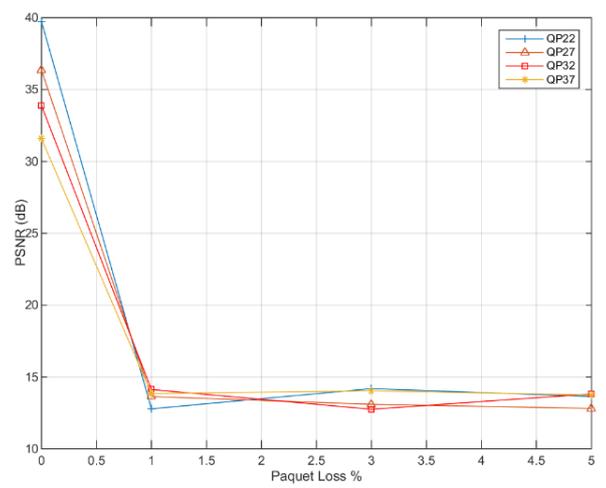
(a)



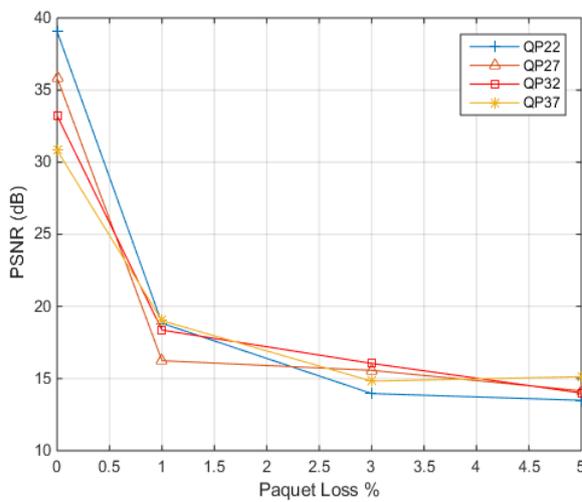
(b)



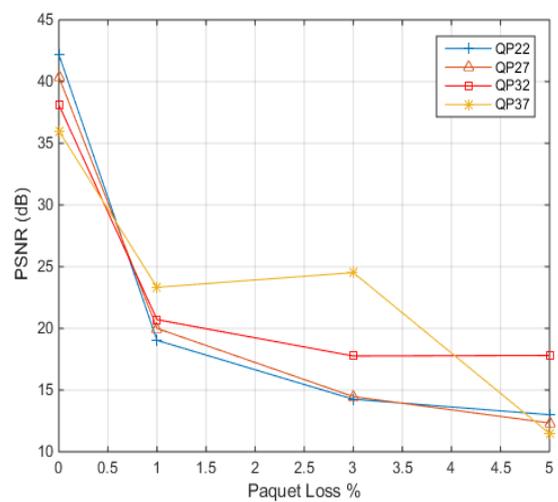
(c)



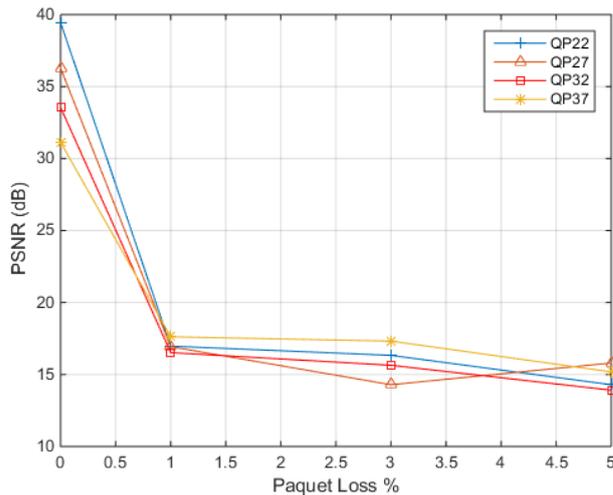
(d)



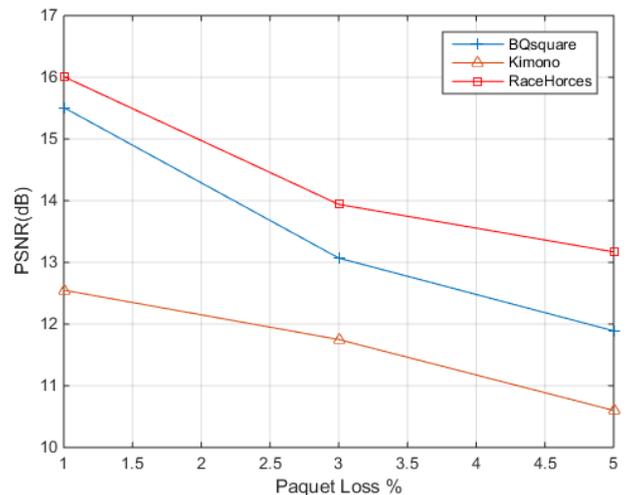
(e)



(f)



(g)



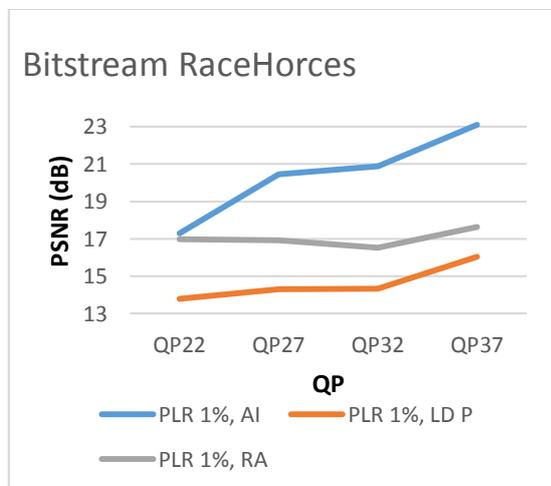
(h)

Figure 3.16 Variation du PSNR en fonction du PLR

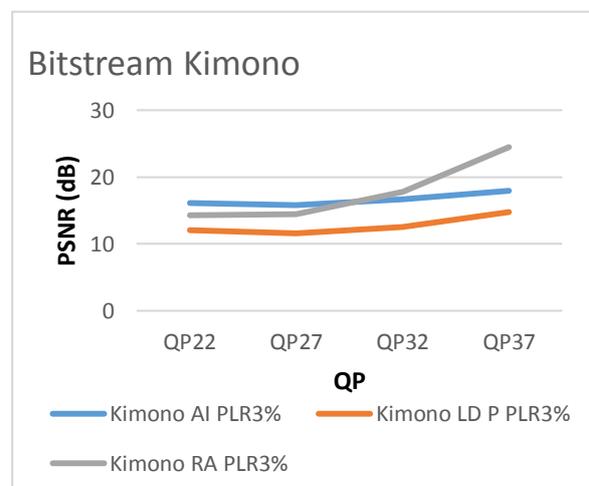
- (a) (BQsquare, AI), (b) (Kimono, AI), (c) (RaceHorses, AI), (d) (BQsquare, LDP),
 (e) (BQsquare, RA), (f) (Kimono, RA), (g) (RaceHorses, RA)
 , (h) ((BQsquare, Kimono, RaceHorses) non compressés)

Pour vérifier l'analyse présentée précédemment, nous avons tracé les courbes des variations du PSNR en fonction des configurations et ceci pour un même PLR.

D'après les figures, On remarque que le PSNR de la configuration All Intra est le meilleur quel que soit le QP, après on aura le PSNR de la configuration Random Access et enfin, le PSNR de la configuration Low Delay P.



(a)



(b)

Figure 3.17 Variation du PSNR en fonction de QP pour différentes configurations

3.5.4 Vérification du caractère aléatoire des pertes de paquets :

Mais, parfois, les résultats ont tendance à s'inverser pour certaines valeurs de QP. C'est le cas de la figure 3.17(b) pour QP= 32 et QP=37 pour les configurations AI et RA. Nous avons alors pensé à tester la caractéristique aléatoire de la perte de paquets produites par l'outil NETEM.

Pour le prouver, nous avons transmis le Bitstream (Kimono, All Intra) trois fois pour montrer la différence entre les tailles des fichiers reçus et leurs PSNRs. Les résultats sont récapitulés dans les deux tableaux suivants :

Tableau 3.13 Vérification du caractère aléatoire de la perte de paquets (Kimono, AI)

Kimono 1920*1080@24 ALL INTRA			QP 22	QP 27	QP 32	QP 37
Taille HEVC transmis			27.8 Mo	15.2 Mo	8.6 Mo	4.8 Mo
HEVC reçu PLR 1%	Test1	(Taille) (PSNR)	27.512 18.67	15.044 21.34	8.456 20.88	4.742 24.54
	Test2	(Taille) (PSNR)	27.490 19.10	15.048 20.24	8.466 21.89	4.704 22.54
	Test3	(Taille) (PSNR)	27.56 19.71	15.058 20.95	8.488 20.6	7.722 22.97
HEVC reçu PLR 3%	Test1	(Taille) (PSNR)	26.97 16.13	14.76 15.84	8.286 18.68	4.618 17.97
	Test2	(Taille) (PSNR)	26.819 15.90	14.745 15.88	8.307 18.29	4.634 19.34
	Test3	(Taille) (PSNR)	26.936 14.46	14.735 16.71	8.302 17.38	4.625 19.07
HEVC reçu PLR 5%	Test1	(Taille) (PSNR)	26.341 12.83	14.395 14.20	8.126 15.29	4.533 16.38
	Test2	(Taille) (PSNR)	26.451 13.19	14.457 13.9	8.151 15.51	4.566 16.5
	Test3	(Taille) (PSNR)	26.447 13.32	14.42 15.31	8.089 15.59	4.548 17.73

On constate alors qu'on a une perte/gain de ± 1 Mo dans la taille du fichier reçu et une variation de ± 2 dB dans la qualité, et cela à cause de la nature aléatoire des pertes de paquets produites par NETEM.

3.5.5 Vérification des valeurs des PSNR :

Comme mentionné précédemment, FFmpeg a été utilisé pour évaluer la qualité des vidéos en utilisant le PSNR comme métrique, et comme nous avons constaté des chutes et des dégradations importantes du PSNR, on a pensé à vérifier et confirmer ces résultats par d'autres outils tels que MATLAB, VQMT et la plateforme HM. Les résultats sont présentés dans les tableaux suivants :

Tableau 3.14 (Kimono, All Intra) résultats

PSNR avec différents outils

Kimono1		QP22	QP27	QP32	QP37
ALL INTRA					
PSNR [db]	FFmpeg	43.34	41.52	39.55	37.37
	Matlab	44.12	42.21	40.43	38.68
	VQMT	42.9	41.1	39	36.6
	HM	43.52	41.7	39.83	37.88

Tableau3.15 (RaceHorses, LD P) résultats

PSNR avec différents outils

RaceHorses		QP22	QP27	QP32	QP37
Low Delay P					
PSNR [db]	FFmpeg	40.27	36.58	33.64	31.03
	Matlab	40.98	37.74	35.15	32.89
	VQMT	40.65	36.68	33.35	30.34
	HM	40.56	37.22	34.5	32.18

3.5.6 L'effet de la perte de paquets sur la vidéo :

Les figures 3.18, 3.19, et 3.20, montre que le PLR peut provoquer des blocs dans la vidéo décodée



PLR 0%

PLR 1%



PLR3%

PLR5%

Figure 3.18 Effet de PLR sur la vidéo (Kimono, Low Delay P, QP37)



PLR 0%

PLR 1%

Figure 3.19 Effet de PLR sur la vidéo (BQsquare, ALL INTRA, QP22)



PLR 0%

PLR 3%

Figure 3.20 Effet de PLR sur la vidéo (RaceHorses, Random Access, QP32)

3.6 CONCLUSION :

Dans ce chapitre, nous avons vu la configuration et les outils utilisés pour assurer une transmission sans fil. Le logiciel FFMPEG a été exploité pour la transmission et la réception des bitstreams. Ces derniers ont été produits par le standard HEVC et l'outil NETEM pour générer les différents taux de pertes de paquet.

D'autre part, nous avons vu que NETEM génère des pertes aléatoires. C'est ce qui explique que pour chaque transmission nous avons des résultats différents (taille perdue et PNSR). Cependant, cette perte est toujours proche du PLR utilisé. La perte est évidemment d'autant plus élevée que le PLR augmente. Ces altérations des fichiers reçus influent énormément sur la qualité de la vidéo, même un PLR très petit provoque une dégradation totale du PSNR.

Nous avons aussi remarqué que les pertes affectent les trois configurations d'une manière différente où le plus touché est la vidéo compressée avec une configuration de type LDP.

Conclusion générale :

Le monde de multimédia et de télécommunication a évolué à une vitesse vertigineuse ces dernières années. Les capacités de traitement et de stockage de tous les types d'appareils (ordinateurs, smartphones, tablettes, ...) ne cessent d'augmenter. D'autre part, la transmission vidéo sur les réseaux IP qui permet de rendre le streaming vidéo réalisable est confrontée au dilemme qualité nécessitant des débits importants et bandes passantes limitées. La seule solution est la compression. Cependant, les canaux de transmission sont très hostiles, particulièrement pour les vidéos compressées, qui ont perdu une certaine quantité d'information. L'un des problèmes les plus importants est la perte de paquets.

L'objectif de cette thèse est de transmettre des vidéos codées par la nouvelle norme H.265/HEVC sur un réseau sans fils Ad-hoc et d'évaluer l'impact des pertes de paquets sur la qualité.

On estime que l'objectif principal a été atteint. En utilisant l'outil FFmpeg et un émulateur de réseaux "NETEM", les "bitstream" générés par la norme HEVC (en utilisant la plateforme HM 16.11) ont été transmis sur un support sans fils. Le taux de perte de paquet "PLR" a été contrôlé et a varié de 1% à 5%. Puis la qualité des vidéos reçues a été mesurée objectivement par le calcul du PSNR. Au-delà de 5%, la dégradation est totale.

Ce travail a démontré que la moindre perte de paquets (faible PLR) entraîne des dégradations plus ou moins importantes sur les vidéos. Un choix judicieux du taux de compression et de la configuration permettrait de réduire ces dommages. En effet, les résultats ont prouvé que les frames codés en Intra et qui sont des frames clés sont les plus sensibles aux pertes. Par conséquent, le choix d'une configuration à grand nombre de frames I résulterait en une meilleure qualité. C'est ce qui permet de conclure que le mode de compression ALL INTRA donnerait les meilleurs résultats.

Enfin, le codec HEVC, très efficace en compression est incapable de décoder une vidéo qui a subi des pertes. Ce problème a été résolu par l'utilisation du FFmpeg dont a découvert toute la puissance dans le traitement des vidéos.

Dans ce projet, on a aussi été amené à exploiter d'autres logiciels tels que Matlab et VQMT pour s'assurer des valeurs d'évaluation de la qualité.

Comme perspectives, nous proposons une suite à ce travail qui consiste en :

- ✚ Utilisation d'autres configurations de réseaux : Plusieurs PC, avec routeurs, ... etc.
- ✚ Tester la transmission avec d'autres protocoles tels que
- ✚ Tester la transmission des frames clés d'une vidéo avec un protocole de transmission et une compression légère. Les autres frames peuvent être compressés avec un taux plus important.

Références bibliographiques

- [1] L'encyclopédie libre Wikipédia https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio
- [2] <http://www.rennes.supelec.fr/ren/perso/jweiss/tv/perception/qualite.php>
- [3] L'encyclopédie libre Wikipédia
https://fr.wikipedia.org/wiki/Codec#Codecs,_normes_et_conteneurs
- [4] L'encyclopédie libre Wikipédia https://en.wikipedia.org/wiki/Video_codec#Applications
- [5] L'encyclopédie libre Wikipédia https://en.wikipedia.org/wiki/H.264/MPEG-4_AVC#cite_note-2
- [6] L'encyclopédie libre Wikipédia <https://fr.wikipedia.org/wiki/H.265/HEVC>
- [7] Mohd.A.Saleh “Review for High Efficiency Video Coding (HEVC)”, December 2014
- [8] L'encyclopédie libre Wikipédia https://fr.wikipedia.org/wiki/Mod%C3%A8le_OSI
- [9] Guy Pujolle, Initiation aux réseaux, Editions Eyrolles 61, Bld Saint-Germain 75240 Paris cedex 05, www.editions-eyrolles.com, Éditeur délégué : Olivier Salvatori
- [10] Guy Pujolle, “Cours réseaux et télécoms Avec exercices corrigés,” Avec la contribution de Olivier Salvatori, 3 e édition © Groupe Eyrolles, (2008).
- [11] L'encyclopédie libre Wikipédia https://en.wikipedia.org/wiki/Real-time_Transport_Protocol#Packet_header
- [12] L'encyclopédie libre Wikipédia https://en.wikipedia.org/wiki/Internet_Protocol#Function
- [13] https://www.memoireonline.com/01/09/1878/m_Les-technologies-sans-fil-Le-routage-dans-les-reseaux-ad-hoc-OLSR-et-AODV2.html
- [14] L'encyclopédie libre Wikipédia
https://en.wikipedia.org/wiki/Network_simulation#Simulations
- [15] L'encyclopédie libre Wikipédia
https://en.wikipedia.org/wiki/Network_emulation#Emulation,_simulation_and_traffic_generation

[16]A. Salima épouse BRAHAMI Evaluation du Protocole Directed Diffusion dans un réseau decapteurs sans fil mémoire master en Informatique,Université Abou BakrBelkaid-Tlemcen.Année : 2013-2014.

[17]M. Patil, Rajashekhar C. Biradar" A Survey on Routing Protocols in Wireless Sensor Networks"Bangalore-560 064, India, IEEE 2012.

[18] https://www.hackplayers.com/2015/10/10-emuladores-wan-para-que-pruebes.html?m=1&fbclid=IwAR0yiXT9IhOHvJ6DL2Xbko-yp0WSO6M0ojx7qKe_gvOAIsv21haizOgII1s

[19] L'encyclopédie libre Wikipédia
https://en.wikipedia.org/wiki/Graphical_Network_Simulator-3

[20] Common test conditions and software reference configurations, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Document : JCTVC-L1100, 2013.