

MA-004-435-1



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET
POPULAIRE



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITÉ SAAD-DAHLEB-BLIDA

MÉMOIRE DE MASTER

DISCIPLINE : Informatique
OPTION : Ingénierie de Logiciel

Thème

Développement d'une Plateforme d'évaluation de la Qualité

d'Expérience des Utilisateurs web :

Cas des Services de Streaming vidéo.



Réalisé par

Mansour Hadjer
Trifi Hayet

Promotrice : Mme Aroussi Sana
Présidente : Mme Mancer
Examinatrice : Mme Ghribi

Promotion : 2016-2017

MA-004-435-1

Remerciement

Tout d'abord, nous tenons rendre grâce à dieu tout puissant pour nous avoir donné le courage et la détermination nécessaire pour finaliser ce travail et le mener à terme.

Nulle œuvre n'est exaltante que celle réalisée avec le soutien moral et financier des personnes qui nous sont proches.

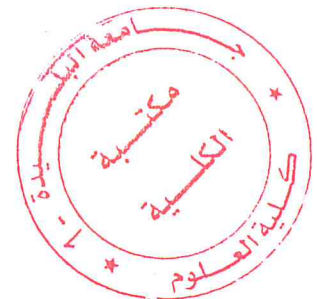
On tient à exprimer nos plus profondes reconnaissances à Nos parents respectifs qui, par leur amour et leur affection nous ont permis d'arriver là où nous sommes aujourd'hui.

Nous remercions notre promotrice Mme AROUSSI qui a endossé son rôle de la meilleure façon qui soit. Nous retiendrons sa patience, sa disponibilité et sa compréhensibilité ainsi que son aide précieuse, ces conseils avisés et ces idées riches.

Nous remercions le membre de jury pour nous avoir fait l'honneur de juger notre travail.

Nous tenons à remercier également et énormément nos amis ainsi toute personne qui nous a aidés de près ou de loin.

Merci.



Dédicace

*C'est avec un immense plaisir que je dédie ce travail
A mes très chers parents qui sont toutes ma vie et tout ce que j'ai de
plus cher au monde, en
témoignage de ma reconnaissance infinie pour ses nombreux sacrifices.
Qu'ils trouvent en ce travail la preuve de mon éternel amour et ma
reconnaissance envers eux.*

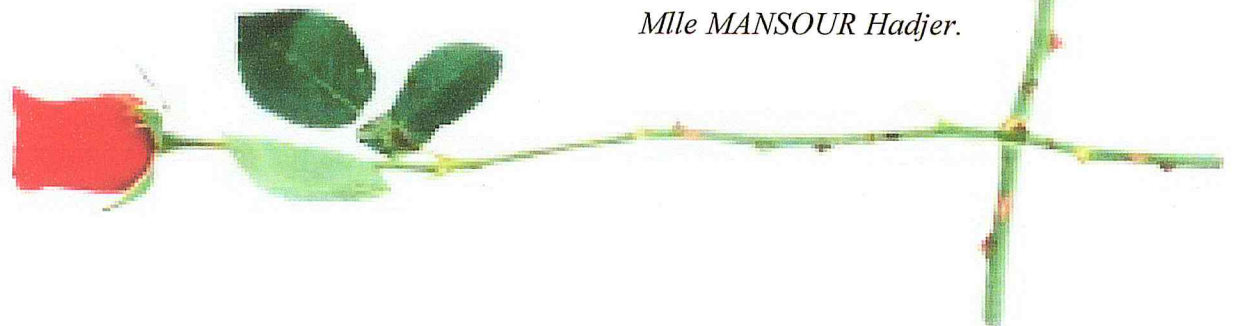
Que dieu les gardes et leur procure la santé et le bonheur.

Aussi à mes

*Sœurs, mes frères, mes belles sœurs, à mes nièces Rodaina & Tassnime,
mes Neveux, mes cousines et toute ma familles.*

Sans oublier Toutes Mes ami(e)s.

Mlle MANSOUR Hadjer.



Dédicace

C'est avec un immense plaisir que je dédie ce travail

A mes très chers parents qui sont toutes ma vie et tout ce que j'ai de

plus cher au monde, en

témoignage de ma reconnaissance infinie pour ses nombreux sacrifices.

Qu'ils trouvent en ce travail la preuve de mon éternel amour et ma

reconnaissance envers eux.

Que dieu les gardes et leur procure la santé et le bonheur.

Aussi à mes

Sœurs, mes frères, mes belles sœurs,

mes Neveux, mes cousines et toute ma familles.

Sans oublier Toutes Mes ami(e)s.

Mlle Trifi Hayet



Résumé

Au cours de ces dernières années, l'explosion d'Internet a donné naissance à de nombreux services de streaming vidéo comme la télévision par internet, la vidéo à la demande ou la vidéoconférence. L'acceptabilité et le succès de ces nouveaux services sont conditionnés par le degré de satisfaction de leurs utilisateurs finaux qui est déterminé par la Qualité d'Expérience (Quality of Experience, QoE). Les fournisseurs de ces services vidéo sont appelés à fournir une bonne QoE s'ils veulent rester en course. Pour savoir quels ajustements devraient être apportés et quand les appliquer, il est ainsi nécessaire d'évaluer la QoE. Or, cette QoE peut être affectée par plusieurs facteurs de natures différentes, à savoir : les caractéristiques du contenu source, les paramètres du codec (compression/décompression), les paramètres de la qualité du service du réseau (bande passante, taux de perte, délai, gigue...), le type du terminal (smartphone, ordinateur portable, tablette, PDA et TV), le profil de l'utilisateur (âge, sexe, compétence visuelle, expérience avec le multimédia ...) et d'autres.

Dans la littérature, plusieurs travaux de recherche ont été menés afin d'étudier l'impact d'un ou de plusieurs de ces facteurs sur la QoE. Pour ce faire, il a été toujours nécessaire de réaliser des expériences dans un environnement contrôlé pour construire une base (ou un échantillon) de données contenant les facteurs influant la QoE ainsi que l'évaluation de la qualité par des sujets humains. Notre travail consiste à concevoir une plateforme générique, en environnement contrôlé et réel, permettant aux chercheurs de mener des expériences pour évaluer la QoE à travers de plusieurs facteurs, de générer une base de données avec les résultats récoltés et enfin d'analyser les modèles de prédiction de QoE.

Mots – clés : Streaming vidéo, Qualité d'Expérience (QoE), Plateforme d'expérimentation.

ABSTRACT

In recent years, the explosion of the Internet has spawned many streaming video services such as Internet TV, video on demand (VOD) or videoconferencing. The acceptability and success of these new services are conditioned by the degree of satisfaction of their end users which is determined by the quality Experience (Quality of Experience, QoE).

The Providers of these services are have to provide good QoE if they want to stay competitive in technologies providers market. To find out which adjustments should be made and when it should be applied, it's necessary to evaluate QoE. However, this QoE can be affected by several factors of various types, to namely, the characteristics of the source material, the codec settings (compression / decompression), the parameters of the quality of network service (band bandwidth, loss rate, delay, jitter ...), the type of terminal (smartphone, computer Laptop, Notebook, PDA and TV), the user's profile (age, gender, visual competence, experience with multimedia ...) and other.

In the literature, several works Research was conducted to study the impact of one or more of these factors on QoE. To do this, it is always necessary to carry out experiments in a controlled environment to build a base (or a sample) of data containing the factors affecting the QoE and the quality assessment by humans subjects. This topic is to design a generic platform, in both controlled and real environment, allowing researchers to conduct experiments to assess the QoE through several factors, generate a database with the results collected and finally analyze QoE prediction models.

Keywords: Streaming video, Quality of Experience (QoE), Platform for experimentation.

ملخص

في السنوات الأخيرة، ولد تطور الإنترنت العديد من خدمات الفيديو مثل تلفزيون الإنترنت، والفيديو حسب الطلب (فود) أو عقد المؤتمرات بالفيديو. ويتوقف مقبولية ونجاح هذه الخدمات الجديدة على درجة رضا المستخدمين النهائيين التي تحددها نوعية الخبرة (نوعية الخبرة، كو).

يجب على مقدمي هذه الخدمات توفير نوعية جيدة من كو إذا كانوا يرغبون في البقاء على المنافسة في سوق مزودي التقنيات و لمعرفة ما هي التعديلات التي يجب إجراؤها ومتى يجب تطبيقها، من الضروري تقييم جودة. ومع ذلك، هذا كو يمكن أن تتأثر بعدة عوامل من أنواع مختلفة، أي خصائص المواد المصدر، وإعدادات الترميز (ضغط / الضغط)، ومعلمات نوعية خدمة الشبكة (عرض النطاق الترددي الفرقة، ومعدل الخسارة، والتأخير، غضب ...)، ونوع من محطة (الهاتف الذكي، الكمبيوتر المحمول، دفتر، المساعد الشخصي الرقمي والتلفزيون)، الملف الشخصي للمستخدم (العمر والجنس والكفاءة البصرية، والخبرة مع الوسائط المتعددة...) وغيرها. في الأدب، تم إجراء العديد من الأبحاث لدراسة تأثير واحد أو أكثر من هذه العوامل على كو. للقيام بذلك، من الضروري دائما لإجراء تجارب في بيئة تسيطر عليها لبناء قاعدة (أو عينة) من البيانات التي تحتوي على العوامل التي تؤثر على جودة كو وتقييم الجودة من قبل البشر الموضوعات. هذا الموضوع هو تصميم منصة عامة، في كل من البيئة التي تسيطر عليها والحقيقية، مما يسمح للباحثين لإجراء تجارب لتقييم كو من خلال عدة عوامل، إنشاء قاعدة بيانات مع النتائج التي تم جمعها وأخيرا تحليل نماذج التنبؤ كو.

الكلمات: الفيديو , (QoE) جودة التجربة , منصة للتجريب

Table des matières

INTRODUCTION GENERALE.....	1
Partie I : ETUDE BIBLIOGRAPHIQUE.....	
Chapitre I : QoE des Services de Streaming Vidéo.....	4
I.1 Introduction.....	4
I.2 Les services de Streaming Vidéo.....	5
I.2.1 Définitions.....	5
I.2.2 Les différents types de Streaming vidéo.....	5
I.2.3 L'architecture de Streaming à la demande.....	
Erreur ! Signet non défini.	
I.2.4 Les protocoles de transmission de Streaming à la demande.....	8
I.3 La Qualité d'Expérience.....	9
I.3.1 Définitions.....	9
I.3.2 Modèle de Métriques de la QoE.....	10
I.3.3 Méthodes de Mesures.....	22
I.3.4 Modèle de Prédiction.....	23
I.4 Conclusion.....	24
Chapitre II : Travaux Antérieurs.....	25
II.1 Introduction.....	25
II.2 Définitions.....	25
II.3 Environnement de test.....	25
II.3.1 Environnement Contrôlé.....	26
II.3.2 Environnement Réel.....	26
II.3.3 Comparaison.....	27
II.4 Architecture générale de la plateforme d'expérimentation.....	27
II.5 Principales Plateformes existantes.....	27
II.5.1 Plateforme de LAGHARI (environnement Contrôlé).....	28

II.5.2 Plateforme de BENYOUTTOU	29
II.5.3 Plateforme de LAGHARI (environnement Réel)	30
II.5.4 Plateforme de Musthaq et al.....	32
II.5.5 Plateforme d'Amour	34
II.6 Conclusion	39
Partie II : Conception et implementation.....	
Chapitre III : Etude Conceptuelle	40
III.1 Introduction.....	40
III.2 Présentaion de la démarche utilisée	40
III.2.1 Le cycle de vie	40
III.3 Le modèle en Cascade	40
III.4 Expression des besoins des besoins.....	42
III.4.1 Besoins Foncionnelles	42
III.4.2 Besoins non Fonctionnelles	42
III.5 Les diagrammes de cas d'utilisations	43
III.5.1 Gestion des expériences.....	45
III.5.2 Gestion de réseau local	47
III.5.3 Statistiques.....	48
III.6 Les diagrammes de séquences	49
III.7 L'architecture de la plateforme.....	50
III.8 Notre Modèle de Métrique.....	52
III.8.1 Le profil de testeur.....	53
III.8.2 Les paramètres de terminal.....	53
III.8.3 Les caractéristiques de la vidéo	54
III.8.4 Les paramètres lieés au paramètre de la qaulité de service (QoS)	55
III.9 Schéma relationnel de la BDD	55
III.9.1 Diagramme de classes	56

III.9.2 Description des classes	57
III.9.3 dictionnaire des données des relations	60
III.9.4 Passage au Modèle relationnel.....	60
III.9.5Le Schéma relationnel	61
III.10 Le processus de collecte d'informations.....	62
III.10 Conclusion	65
Chapitre IV : implemntation.....	66
IV.1 Introduction	66
IV.2 Environnement de developpement.....	66
IV.2.1Outils de developpement	67
IV.3 Présentation de la plateforme « QoEStreamingVideo»	75
IV.3.1Espace administrateur.....	75
IV.3.2Espace client.....	79
IV.4 Conclusion.....	83
Conclusion générale	82
Reference Bibliographique	

Liste des figures

Figure 1 : L'organisation du mémoire.....	3
Figure 2 : Les différents types de diffusion en temps réel (4).....	5
Figure 3: La diffusion d'une vidéo Streaming sur Internet.....	7
Figure 4 : La structure de la QoE (12).....	10
Figure 5: La Structure de la QoE pour le modèle DSL Forum.....	11
Figure 6: Concept multidimensionnel de la QoE (14).....	12
Figure 7: Les facteurs influant la QoE dans le modèle de Diallo.....	13
Figure 8: Facteurs d'Influence du Chen et al (16).	14
Figure 9: Le modèle de métrique de Musthaq.....	16
Figure 10: Le modèle de métrique d'Amour (19).....	17
Figure 11: Catégorisation des modèles de corrélation QoE-QoS (20).	23
Figure 12 : Architecture globale de la plateforme d'expérimentation.	28
Figure 13: Architecture de l'expérience (26).	28
Figure 14: Plateforme de Benyettou (28).	29
Figure 15: Plateforme QoM (26).	31
Figure 16: L'interface graphique de la plateforme QoM (26).	31
Figure 17: L'architecture de la plateforme de Musthaq et al (17).	32
Figure 18: Framework Implémentation (17).	33
Figure 19: Formulaire de commentaire de l'utilisateur (17).	33
Figure 20: Plateforme CLIF (18).....	34
Figure 21: Capture d'écran de vidéos utilisées (18).....	34
Figure 22 : Des captures d'ecran sur les interfaces de la plateforme (18).	35
Figure 23 : Cycle de vie selon le modèle en cascade (28).....	41
Figure 24: Description du diagramme Global de cas d'utilisation.....	44
Figure 25 : Diagramme de cas d'utilisation « gestion des expériences ».	46
Figure 26 : Diagramme de cas d'utilisation" Gestion du réseau local ".....	47
Figure 27: Diagramme de cas d'utilisation "Statistiques".....	48
Figure 28: Diagramme de séquence de cas d'utilisation " ajout d'une expérience".....	49
Figure 29: Diagramme de séquence de cas d'utilisation " test d'une expérience".	50
Figure 30: Diagramme de séquence de cas d'utilisation "Statistiques".	51
Figure 31: Architecture de la plateforme en environnement Controlé.	52

Figure 32: Diagramme d'Ishikiawa de notre modèle de métriques.	53
Figure 33 : Diagramme de classe de la plateforme.....	56
Figure 34: Le schéma relationnel de la base des données.	59
Figure 35: Processus de collecte d'information cas " environnement réel".....	61
Figure 36: Processus de collecte d'information cas "environnement contrôlé".....	62
Figure 37 : L'architecture globale de la plateforme QoEStreamingVideo	66
Figure 38: l'installation de serveur apache2 sous ubuntu.	67
Figure 39: l'installation de serveur MySQL sous ubuntu	67
Figure 40: l'installation de PhpMyAdmin sous ubuntu.	67
Figure 41: l'installation de PHP sous ubuntu.....	68
Figure 42: l'installation de FFmpeg sous ubuntu.....	69
Figure 43: Exemple d'un fichier de configuration de ffmpeg.....	70
Figure 44: le lancement de ffmpeg.	70
Figure 45: l'installation de VLC sous ubuntu.	70
Figure 46: la configuration d'un flux de réseau du lecteur VLC.	71
Figure 47: L'intégration de VLC avec Php	71
Figure 48: l'installation de Médiainfo sous ubuntu.	71
Figure 49: Le résultat obtenu avec Médiainfo sous ubuntu.	72
Figure 50: Collecter les informations d'un vidéo avec mediainfo.	72
Figure 51: L'intégration de médiainfo sous php.....	72
Figure 52: l'installation d'INXI sous ubuntu.	73
Figure 53: le résultat obtenu avec inxi.....	73
Figure 54: exemple commande Netem sur les paquets émis.....	74
Figure 55: Exemple commande Netem sur les paquets reçus.	74
Figure 56: L'interface d'authentification.....	75
Figure 57: Accueil administrateur.	76
Figure 58: Gestion de configuration de réseau.....	76
Figure 59: Gestion des expériences.....	77
Figure 60: Ajout d'une nouvelle expérience.....	78
Figure 61: Liste des vidéos disponibles.....	78
Figure 62: Formulaire d'inscription.....	79
Figure 63: Accueil Client.....	80
Figure 64: Liste des expériences disponibles.....	80
Figure 65: Liste des vidéos disponibles d'une expérience choisie.....	81

- Interfaçage avec des logiciels PMS (Propriety Management System) qui permettent une facturation simple, souple et automatisée du service
- Possibilité de personnalisation des interfaces graphiques utilisateur

1.2.4 Les protocoles de transmission du streaming à la demande

Les technologies de diffusion vidéo utilisées sur le réseau peuvent être classées en fonction de leur protocole de transport. Nous pouvons distinguer (7):

- **Streaming sur TCP** : TCP est un protocole fiable en mode connecté qui livre un flux de données dans le même ordre que celui dans lequel il a été envoyé, après avoir d'abord établi une session de communication. De nos jours, la diffusion vidéo sur Internet est habituellement basée sur TCP en raison de sa fiabilité. Bien que TCP soit actuellement le protocole de transport le plus utilisé dans Internet, il est généralement considéré comme inapproprié pour la diffusion multimédia pour les raisons suivantes : 1) le mécanisme d'accusé de réception du TCP qui ne se balance pas bien lorsque le nombre des destinations augmente pour le même serveur, et 2) le mécanisme de retransmission qui peut entraîner des retards de transmission indésirables et peut nuire aux exigences de temps strictes pour les flux diffusés vidéo en direct. Afin de faciliter la traversée des NAT et des pare-feu, Hypertext Transfer Protocol (HTTP) a généralement été utilisé sur TCP. En conséquence, le lecteur vidéo est incorporé dans un navigateur Web. Pour la diffusion de vidéos HTTP / TCP, on trouve le protocole HAS (HTTP Adaptive Streaming (HAS)) qui consiste à stocker des contenus vidéo sur un serveur Web, appelé serveur HTTP et diviser le fichier d'origine en segments indépendants de même durée, appelés "morceaux". Chaque morceau est transcodé en différents taux d'encodage inférieurs. Le joueur HAS, du côté client, demande des morceaux, décode et les affiche successivement sur l'écran du client. La spécificité de la technologie HAS est qu'elle offre une adaptabilité aux conditions du réseau; En fait, il permet au client HAS de passer d'un niveau de qualité à l'autre dans le même flux HAS. De nombreux produits commerciaux de HAS ont été proposés, tels que Microsoft Smooth Streaming (MSS), Apple HTTP Live Streaming (HLS) et Adobe HTTP Dynamic Diffusion.

- **Streaming sur UDP** : Un grand nombre de services de diffusion vidéo utilise UDP comme protocole de transport. En fait, UDP est un simple protocole de transmission sans connexion sans dialogues et n'offre pas la garantie de livraison, de commande ou de retransmission. Ces caractéristiques simplifient transmission de données et réduction du délai de bout en bout. Par conséquent, UDP est adapté pour les applications du temps réel, telles que la diffusion vidéo en direct, qui sont sensibles aux retards. Il est largement utilisé avec les protocoles de couche d'application suivants :

- **RTP (Real-Time Transfer Protocol)**

Standardisé en 1996, permet la transmission de données en temps réel. RTP fournit très peu de correction d'erreurs : les paquets perdus, en retard ou endommagés sont ignorés. En effet, il privilégie l'enchaînement du son et des images, plutôt que l'intégrité des données. Cependant, si la vitesse de connexion est inférieure au débit de données, la lecture est "saccadée" voire

Figure 66: Lancement du test.	81
Figure 67: Evaluation de la QoE.	82
Figure 68: Partie statistiques.....	82
Figure 69: Résultat.....	83
Figure 70: Extraction des données.....	83

Liste des tableaux

Tableau 1 : Comparaison entre les différents types de Streaming Vidéo.....	6
Tableau 2: Tableau comparatif entre les modèles de métriques.....	20
Tableau 3: Mean Opinion Score (MOS).....	22
Tableau 4: L'environnement contrôlé vs l'environnement réel.....	27
Tableau 5 : Combinaisons d'ensemble de données (28).....	30
Tableau 6 : Variation de QoS (18).....	35
Tableau 7 : Tableau comparatif entre les plateformes.....	37
Tableau 8: Description du diagramme global de cas d'utilisation.	43
Tableau 9: Description du cas d'utilisation " Gestion des expériences".....	44
Tableau 10: Description du cas d'utilisation " Gestion des expériences".....	45
Tableau 11: Description du cas d'utilisation « Statistiques ».	46
Tableau 12 : Tableau descriptif des classes	59
Tableau 13 : Dictionnaire des données des relations :.....	60
Tableau 14: FIs collectées à chaque étape.....	62

Liste des abréviations

AAC : Advanced Audio Coding,
ACR : Absolute Category Rating
AQ : Audio Quality
FR : Full Reference
FTP : File Transfer Protocol
HTTP : HyperText Transfer Protocol
IP : Internet Protocol
ITU-T : ITU Telecommunication Standardization Sector
IPTV : IP TeleVision
ITU : International Telecommunication Union
MOS : Mean Opinion Score
MPQM : Moving Picture Quality Metric
MSE : Mean Square Error
NR : No Reference,
NVFM : Normalized Video Fidelity Metric
PESQ : Perceptual Evaluation of Speech Quality
PEVQ : Perceptual Evaluation Video Quality
PSNR : Peak Signal to Noise Ratio
PMS : Propriety Management System
QOE : Quality Of Experience
QOS : Quality Of Service
RTP : Real time Transport Protocol
RTCP : Real-time Transport Control Protocol
RTSP : Real Time Streaming Protocol
RR : Reduced Reference
TCP : Transmission Control Protocole
UDP : User Datagram Protocol
VCR : Video Cassette Recording
VOD : Video On Demand
VQ : Video Quality
VQEG : Video Quality Expert Group
3SQM : Single Sided Speech Quality Measure

Introduction générale

Au cours des dernières années, nous pouvons assister à l'énorme croissance des services multimédias, en particulier les services de diffusion vidéo qui ont prévalu dans l'Internet global avec un partage plus important. Dans la technologie de diffusion multimédia, on distingue deux applications principales : la lecture en transit, qui implique le téléchargement d'un fichier, et la transmission multimédia en continu (*media streaming*), qui permet la diffusion d'un contenu multimédia en temps réel, sans enregistrement sur le poste client (par exemple, la transmission d'une émission de radio par Internet). Ces services de streaming vidéo sont en pleine expansion du fait de la valeur ajoutée qu'ils apportent aux utilisateurs, telle que le contenu multimédia, la transmission des données, la mobilité et le coût réduit des appels longs distance. Fournir un accès Internet à ces nouveaux services implique trois acteurs différents: les opérateurs de réseau, les fournisseurs de services et les fournisseurs de contenu. Cet écosystème multipartite survit car ces différents acteurs répondent aux exigences des clients et de la qualité d'expérience (QoE).

Selon la norme UIT-T (1), La Quality of Experience (QoE) se réfère à *«l'acceptabilité globale d'une application ou d'un service, perçue subjectivement par l'utilisateur final»*. De même, Qualinet (2) définit la QoE comme *«le degré de plaisir ou d'ennui de l'utilisateur d'une application ou d'un service. Cela résulte de l'accomplissement de ses attentes en ce qui concerne l'utilité et / ou la jouissance de l'application ou du service en fonction de la personnalité et de l'état actuel de l'utilisateur »*. Par conséquent, la QoE est une mesure subjective et peut varier selon les attentes et le contexte de l'utilisateur. En outre, il s'agit d'un effet global du système bout en bout (client, terminal, réseau, infrastructure de services, etc.) et dépend d'un certain nombre de facteurs qui ne peuvent être simplement mesurés. Il nécessite des tests avec des utilisateurs réels dans un environnement contrôlé ou réel pour estimer correctement la QoE.

Or, la QoE peut être affectée par plusieurs facteurs de natures différentes , à savoir : les caractéristiques du contenu source, les paramètres du codec (compression/décompression), les paramètres de la qualité du service du réseau (bande passante, taux de perte, délai, gigue...), le type du terminal (smartphone, ordinateur portable, tablette, PDA et TV), le profil de l'utilisateur (âge, sexe, compétence visuelle, expérience avec le multimédia ...) et d'autres. Il faut donc d'étudier l'impact d'un ou de plusieurs de ces facteurs sur la QoE afin de pouvoir la contrôler. Cela a fait l'objet de plusieurs travaux de recherche. Comme il n'existe pas des benchmarks ou des échantillons de données disponibles gratuitement, les chercheurs doivent réaliser leurs propres expériences. Donc, ils se trouvent confrontés à des difficultés de mettre en place leur plateforme d'expérimentation qui leur permet de collecter l'échantillon de données contenant les facteurs influant la QoE ainsi que l'évaluation de la qualité par des sujets humains. Ce qui engendre beaucoup de temps et d'effort, voire un coût élevé.

Notre objectif consiste à concevoir une plateforme générique, en environnement contrôlé et réel, permettant aux chercheurs de mener des expériences pour évaluer la QoE à travers de

plusieurs facteurs, de générer une base (ou un échantillon) de données avec les résultats récoltés et enfin d'analyser les modèles de prédiction de QoE.

Pour atteindre cet objectif, nous avons organisé ce présent mémoire en deux parties :

- La première partie sur l'étude bibliographique comprend deux chapitres :
 - Dans le chapitre 1, nous présentons un état de l'art sur la QoE des services de Streaming Vidéo afin de comprendre la problématique à résoudre
 - Dans le chapitre 2, nous présentons des travaux extérieurs et nous analysons les implémentations de ces plateformes d'expérimentation existantes dans la littérature.
- Une deuxième partie sur la conception et l'implémentation de notre plateforme qui prend également deux chapitres :
 - Dans le chapitre 3, nous décrivons notre conception de la plateforme d'expérimentation pour évaluer la QoE des utilisateurs des services de streaming vidéo.
 - Dans le chapitre 4, nous expliquons d'abord l'implémentations de notre plateforme, ensuite, nous effectuons quelques tests.

En conclusion générale, nous synthétisons les principales parties du mémoire, en faisant ressortir les apports de notre travail ainsi que les perspectives prévues pour l'améliorer. La figure 1 ci-dessous schématise la structure du mémoire.

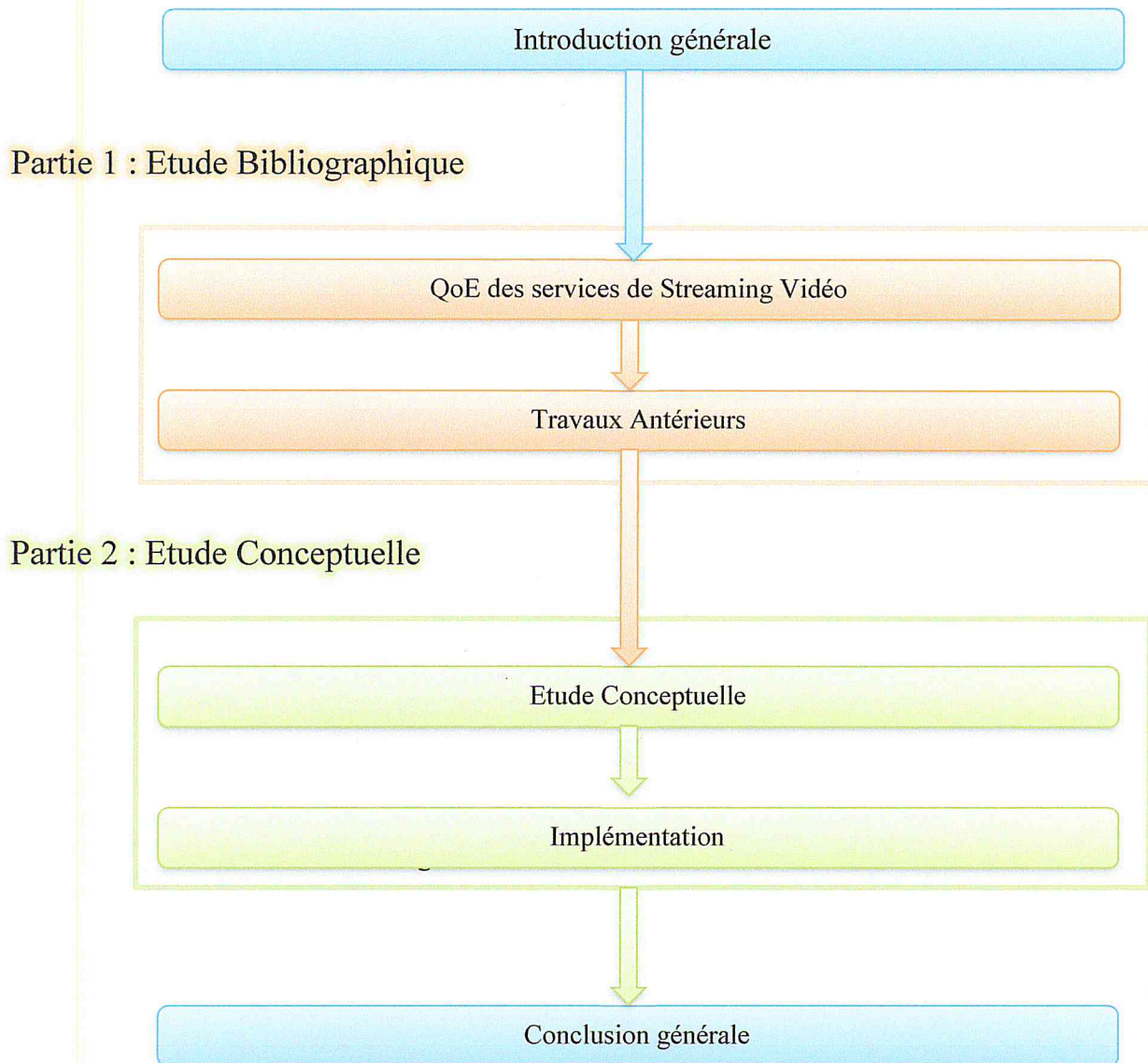


Figure 1 : L'organisation du mémoire.

Partie I

Etude Bibliographique

*« La bibliographie se fait
après et non avant d'aborder
un sujet de recherche »*

Jean Perrin

Chapitre I: QoE des Services de Streaming Vidéo

I.1 Introduction

Les services de streaming sur les réseaux sont en pleine expansion du fait de la valeur ajoutée qu'ils apportent aux utilisateurs, telle que le contenu multimédia, la transmission des données, la mobilité et le coût réduit des appels longs distance. Par ailleurs, la qualité de service (QoS, Quality of Service) de ces nouveaux services est généralement non garantie. Les opérateurs de télécommunications doivent satisfaire leurs clients en offrant la meilleure QoS possible. Donc, ils doivent évaluer et contrôler constamment la qualité perçue par leurs utilisateurs, cette qualité est connue sous le nom de qualité d'expérience (QoE, Quality of Experience) des utilisateurs.

Les méthodes subjectives, nécessitant l'appel à des participants humains pour tester un système, sont le meilleur moyen d'atteindre le jugement des usagers. Bien que ces méthodes subjectives restent la meilleure option pour évaluer la qualité perçue par les utilisateurs ou tout simplement la QoE, le recours à de telles méthodes n'est pas rentable pour les fournisseurs de services et opérateurs de télécommunications, du fait des contraintes temporelles sollicitées. Raisons pour lesquelles, des méthodes objectives ont été développées dans le but d'évaluer la QoE au niveau des utilisateurs et ainsi représentent un moyen de substitution aux mesures subjectives. En effet, en possédant des informations concrètes sur la manière dont la QoE varie en fonction des changements des paramètres de la QoS ou autres, les fournisseurs de services et opérateurs de télécommunications peuvent maintenir la meilleure qualité d'expérience en temps réel.

Ce premier chapitre est divisé en deux parties. Dans la première partie, nous commençons par définir ce qu'un service de streaming, ensuite, nous présentons ses types, son architecture, et les protocoles utilisés pour la diffusion. Dans la deuxième partie, nous étudions la notion de la QoE en citant ses définitions les plus connues, ses différents modèles de métriques, ses méthodes de mesure et enfin ses modèles de prédiction.

I.2 Les services de Streaming Vidéo

Aujourd'hui, plusieurs services de multimédia existent et qui sont utilisés quotidiennement comme la Vidéo à la Demande. Dans cette partie, on va définir le service de streaming, ensuite, nous présentons ses types, son architecture, et les protocoles utilisés pour la diffusion.

I.2.1 Définition

Un Stream est un flux de données de son ou de vidéo transporté par le réseau. Le streaming, synonyme de diffusion de contenus multimédias, est un principe utilisé pour l'envoi de contenus numériques en direct ou indirect (différé après enregistrement). Il permet la lecture d'un flux audio ou vidéo, à mesure qu'il est diffusé. Il s'oppose ainsi à la distribution par téléchargement qui nécessite de récupérer l'ensemble des données composant un contenu, avant de pouvoir l'écouter ou le regarder. Basée sur le principe de la télévision, cette technologie permet de diffuser une information vers un grand nombre d'internautes, mais sans interactivité en direct, sauf si elle est associée à d'autres outils du type « chat », téléphone, ou encore messagerie internet (3).

I.2.2 Les différents types de streaming vidéo

Il y a plusieurs façons pour faire une diffusion en temps réel, comme indiqué dans le schéma ci-dessous (figure 2) (4)

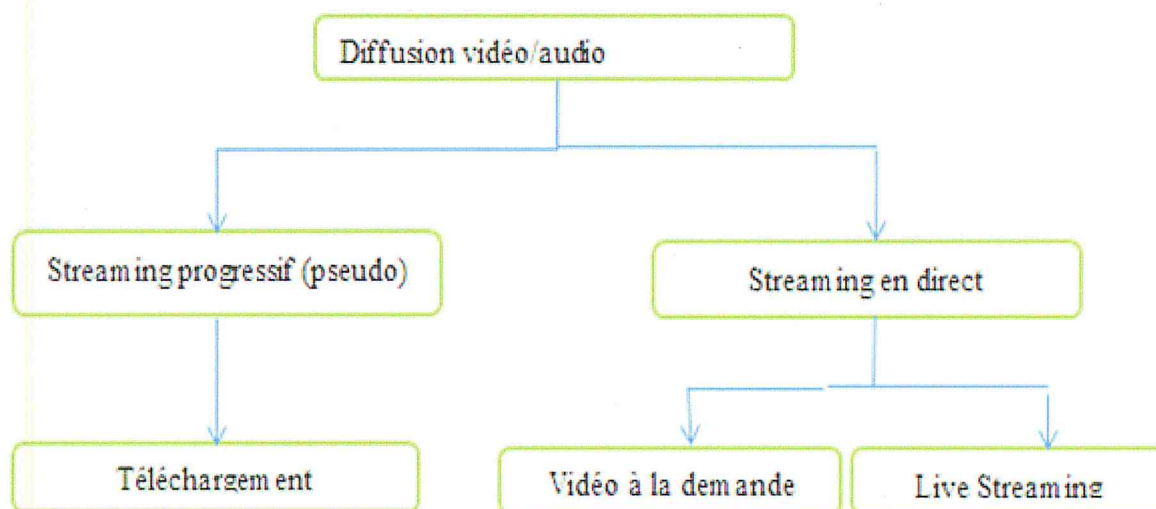


Figure 1 : Les différents types de diffusion en temps réel (4)

Pour le streaming progressif, il oblige l'utilisateur de télécharger le flux (par exemple un film). La reproduction commence déjà pendant le téléchargement, mais on n'a pas la possibilité de gérer le flux, c'est-à-dire d'avancer ou de rembobiner le flux. L'avantage est qu'on peut fournir les données en utilisant un serveur web avec HTTP sans ajouter des fonctionnalités supplémentaires.

Pour le streaming en direct, il présente la vraie diffusion en temps réel, qui peut être faite en ligne (live), ou à la demande :

- Le streaming en ligne permet de diffuser en direct un programme d'une chaîne radio, un cours à l'université ou une conférence audiovisuelle par Internet.
- le streaming à la demande ou VoD (Video on Demand) est utilisée pour diffuser des productions vidéo ou audio précédemment enregistrés, par exemple les chaînes de télévision fournissent des diffusions sélectionnées comme les actualités sur leurs pages web. **Dans notre travail, nous nous intéressons aux services VoD.**

Dans le tableau1 suivant, nous comparons entre ces types de streaming :

Critère de comparaison	Streaming progressif	Streaming en ligne	Streaming à la demande
Stockage de contenu	Le contenu du fichier est stocké dans le dossier temporaire des navigateurs internet	Un seul fichier sur le serveur contenant plusieurs fois les mêmes informations à différents niveaux de qualité.	Stockage (temporaire) des données sur le disque dur local
Début de la lecture	Lancement de la vidéo quand les données sont suffisantes	immédiate	Chargement immédiat du passage que l'utilisateur souhaite voir
Serveur spécialisé	Pas de serveur spécialisé	Nécessite un Serveur spécialisé	Nécessite un Serveur spécialisé
Protocole utilisé	Utilise le protocole http	Utilise les protocoles RTP/RTCP	RTP/UDP ou TCP
Principale avantage	Peu coûteux car pas de serveur spécialisé	Qualité de la vidéo adaptée à la qualité de la connexion	Simple à mettre en œuvre
Principale inconvénient	il ne s'adapte pas la qualité de la vidéo au débit	Plus coûteux à mettre en œuvre	Forte consommation de bande passante

Tableau 1 : Comparaison entre les différents types de Streaming Vidéo.

I.2.3 L'architecture de streaming à la demande

Comme cité précédemment, dans notre travail, nous intéressons au streaming à la demande ou VoD qui est une technique de diffusion unicast de contenus vidéo numériques bidirectionnelle (interactive) offerts ou vendus par les réseaux câblés, comme Internet, ou les réseaux non câblés, comme la téléphonie troisième génération (5). Dans cette section, nous allons présenter l'architecture de ce type de service.

La lecture en continu fonctionne selon le protocole client-serveur. Le contenu est mis à disposition sur un serveur. Le client souhaitant accéder au contenu envoie une requête pour en récupérer une petite partie, à l'endroit du contenu où il souhaite commencer la lecture. La réponse est placée dans une mémoire tampon. Lorsqu'il y a suffisamment de données dans cette mémoire pour permettre de lire le début du fichier audio ou vidéo, la lecture démarre (6) La figure 3 illustre le fonctionnement entier d'une diffusion vidéo à la demande sur Internet comme suit :

1. Via HTTP le client demande un fichier multimédia du serveur web.
2. Le serveur web transmet cette demande au serveur de streaming.
3. Le serveur de streaming établit avec le client une session pour la transmission et la gestion de flux vidéo.
4. Le client regarde les flux vidéo à travers un logiciel de visualisation.

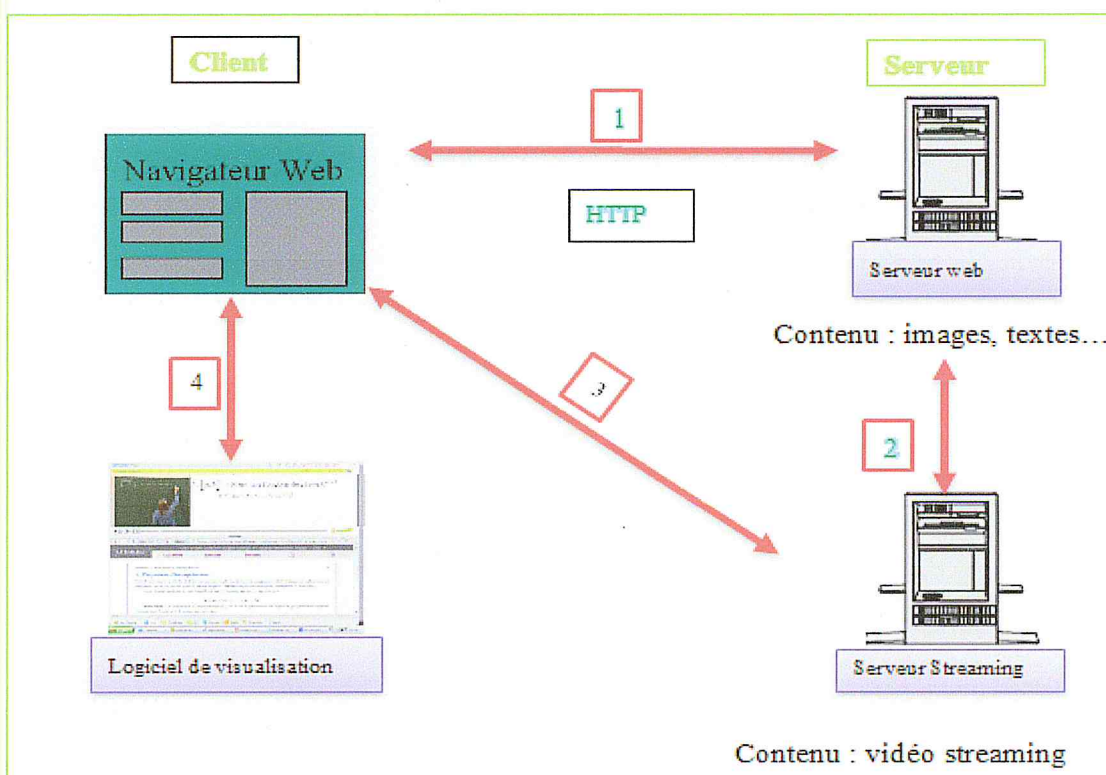


Figure 2: La diffusion d'une vidéo Streaming sur Internet.

Il est à noter que l'application logicielle offre les possibilités suivantes :

- Paramétrage du service pour chacun des utilisateurs (interfaces, contenus, tarifs...)
- Mises à jour fréquentes des contenus audiovisuels disponibles pour les utilisateurs
- Suivi des journaux d'évènements (contrôle distant, liste des achats de films, statistiques...) permettant de renseigner les fournisseurs de contenus

impossible. De même, si le taux d'erreur est supérieur à 10, la lecture est de mauvaise qualité (le son peut par exemple être métallique). Le but de RTP est de fournir un moyen uniforme de transmettre des données soumises à des contraintes de temps réel. Son rôle principal est de mettre en œuvre des numéros de séquence de paquets IP pour reconstituer les infos de voix ou vidéo (3).

- **RTCP (Real Transport Control Protocol)**

Protocole du transport en temps réel est basé sur la transmission périodique de paquets de contrôle à tous les participants d'une session. Ce protocole de contrôle des flux RTP permet de véhiculer des informations basiques sur les participants d'une session, et sur la qualité de service. Le multiplexage des paquets de données RTP et des paquets de contrôle RTCP est réalisé par le protocole sous-jacent (UDP par exemple) (8).

- **RTSP (Real Transport Streaming Protocol)**

Le protocole RTSP (Real-Time Streaming Protocol) est un protocole de contrôle de session pour le streaming média (vidéo/audio) sur Internet. Selon ses développeurs, RTSP fonctionne comme « une commande à distance » du réseau pour les serveurs streaming. A l'instar d'une télécommande de magnétoscope (VCR), il permet aux utilisateurs de demander spécifiquement des données d'un ou de plusieurs serveurs, du type de transfert et d'une destination de transmission des données. En outre, il autorise également les utilisateurs de demander des informations sur les données dans un format spécifique, de lancer, d'arrêter, de mettre en pause la transmission de données, et d'obtenir l'accès direct à différents paquets de données.

Dans notre travail, nous intéressons au streaming à la demande sur UDP.

I.3 Qualité d'Expérience

Avec le développement rapide du marché pour le streaming vidéo d'une année à l'autre, les métriques de la qualité de service comme la bande passante, le délai, la gigue et le taux de perte qui sont généralement utilisés pour garantir les services, ne parviennent pas à mesurer la perception subjective des utilisateurs finaux. Les fournisseurs de ces services ont tendance à muter vers des politiques fondées sur une approche globale de la qualité de bout-en-bout, ainsi est née la qualité de l'expérience (QoE : Quality of Experience). Cette notion fait l'objet de cette partie.

I.3.1 Définitions (9)

Dans la littérature, différents termes font référence à l'expérience utilisateur et plus précisément à la QoE. Les standards pour les transmissions d'ITU (International télécommunication Union) définissent la QoE comme étant « *l'acceptabilité globale d'une application ou d'un service, telle qu'elle est perçue subjectivement par l'utilisateur final. La QoE comprend tous les effets du système global, dont la perception peut être influencée par les attentes de l'utilisateur et le contexte* ».

Martinez-Yelmo et al. proposent quant à eux une définition pour la QoE appropriée à tout domaine : « *il s'agit d'une mesure subjective de l'expérience vécue par l'utilisateur* ». Cette courte définition résume les deux points clés liés au concept de la QoE. Le premier est sous-

jacent au fait que la QoE est basée sur des « mesures », ce qui signifie que certains mécanismes sont nécessaires pour définir quelles mesures sont les plus pertinentes et comment peuvent-elles être obtenues? Le second correspond au fait que « l'expérience est vécue par l'utilisateur », signifiant ainsi que ces mesures sont aussi de nature subjective, font appel à un aspect cognitif et dépendent du profil de l'utilisateur, consommateur de ces services.

Une autre définition plus ample a émergé des discussions faites dans le cadre du séminaire de Dagstuhl 09192 (De la Qualité de Service à la Qualité de l'Expérience) qui détermine la QoE comme « *le degré de satisfaction de l'utilisateur du service* ». Dans le cas de services de communication, elle est influencée par le contenu, le réseau, le dispositif, l'application, les attentes et les objectifs de l'utilisateur ainsi que le contexte dans lequel ils sont utilisés. *Cette définition est la plus utilisée ces dernières années, c'est pourquoi nous la retenons comme définition de la QoE dans ce travail.*

I.3.2 Modèles de métriques de la QoE

L'évaluation de la QoE requiert la définition d'un ensemble de métriques pour mesurer le degré de satisfaction des utilisateurs terminaux (10). Dans la littérature, plusieurs modèles de métriques ont été proposés correspondant à différents types de services. Dans cette section, nous décrivons les modèles les plus importants pour notre travail.

I.3.2.1 Le Modèle d'IUT-T (11)

Selon la norme IUT-T, la QoE mesure la qualité perçue par les utilisateurs finaux en tenant compte de tous les facteurs qui influent à la fois sur leur perception et leurs attentes. Ces dernières sont fortement influencées par plusieurs facteurs subjectifs. La mesure de la perception intègre par ailleurs de multiples paramètres techniques objectifs. Les mesures objectives proviennent d'indicateurs quantitatifs définissant la qualité de service fournie à l'utilisateur en termes de métriques de performance mesurables au niveau du service (la disponibilité), du réseau (le délai, la gigue et le taux de perte) et de l'application client (le codec, la cadence et la résolution d'image). La partie subjective définit quant à elle la qualité perçue par l'utilisateur en termes d'émotions, de facturation du service et d'expérience. La structure de la QoE est illustrée dans la figure4.

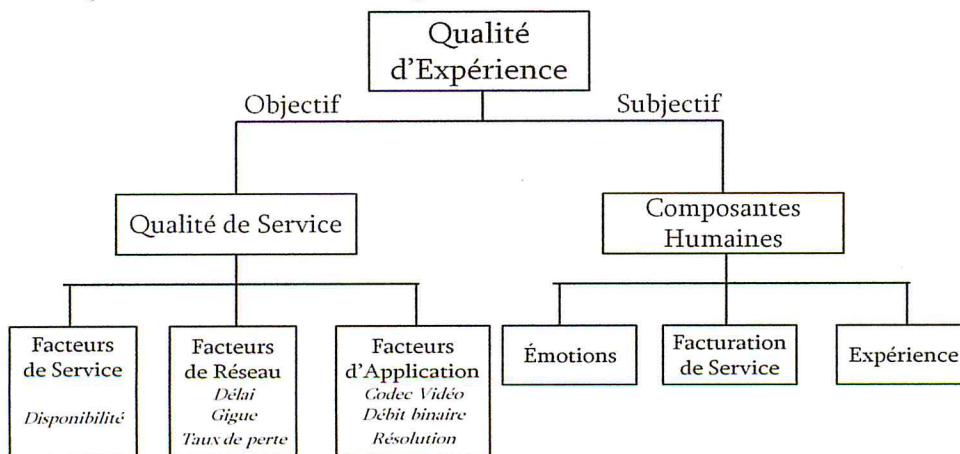


Figure 3 : La structure de la QoE(11).

I.3.2.2 Le Modèle du DSL Forum (12)

A chaque service, le DSL Forum a proposé des facteurs de mesures et des seuils d'acceptabilité (des niveaux prévus) pour la QoE. Dans le cas d'un service vidéo, cette multi-dimensionnalité inclut les facteurs suivants : la fidélité de l'information (la qualité audio et vidéo), la facilité d'utilisation (la manipulation de l'interface de l'utilisateur), l'instantanéité (temps de réponse faible), la sécurité (l'authentification des utilisateurs et la protection des sources vidéo), la fiabilité et la disponibilité. Ces facteurs sont examinés au niveau de trois couches : service, application et réseau. Dans la couche de service, la QoE est mesurée généralement à l'aide de l'échelle MOS (*Mean Opinion Score*). Dans les autres couches, la QoE est corrélée aux paramètres liés à l'application vidéo (la résolution d'image, le codec vidéo/audio, le débit binaire et la couleur) et à la performance du réseau (le délai, la gigue et le taux de perte de paquets). Ces paramètres sont gérés de manière à atteindre les niveaux prévus de la QoE. La structure de la QoE est illustrée dans la figure 5.

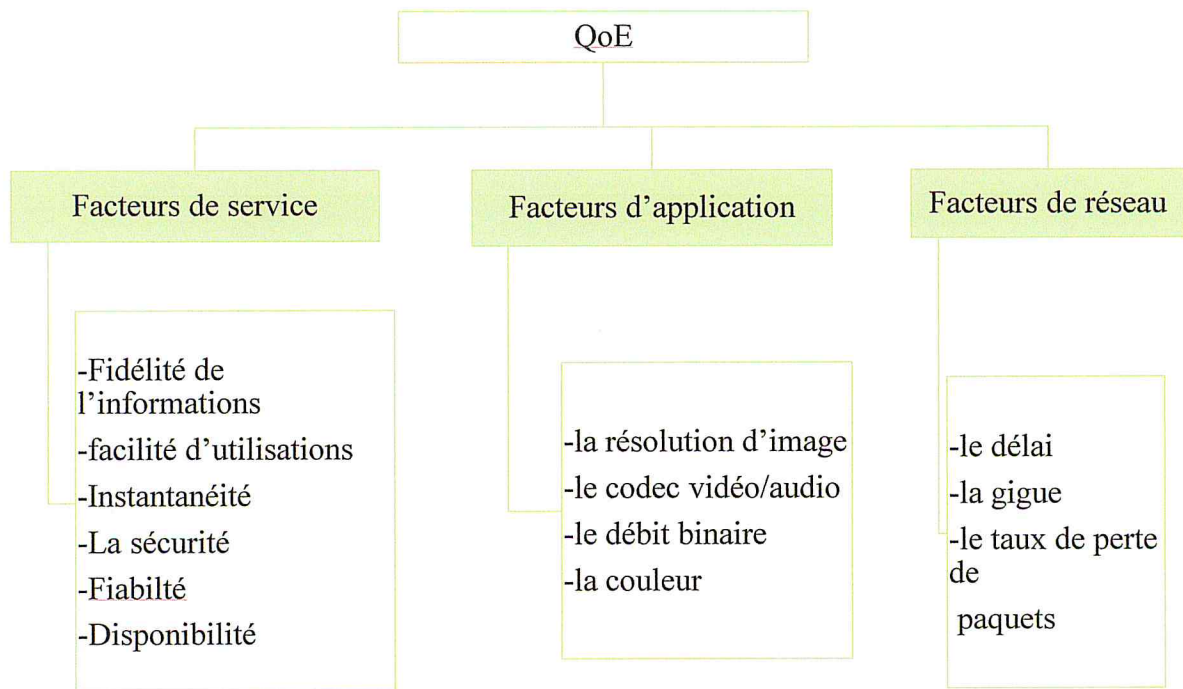


Figure 4: La Structure de la QoE pour le modèle DSL Forum.

I.3.2.3 Le Modèle de Marez & Moor (13)

Dans (14), les auteurs ont proposé un modèle conceptuel couvrant les dimensions suivantes (Figure6):

- La qualité de service qui est une première condition pour parvenir à un service performant. Cette dimension agit sur les performances technologiques à quatre niveaux: application/ service, serveur, réseau et dispositif / appareil.
- La facilité d'usage qui est abordée en termes de l'usage comportementale (basé sur la facilité de travail, la convivialité et l'interaction homme-machine) et de l'usage émotionnel (basé sur les émotions et les sentiments de l'utilisateur lors de l'utilisation du dispositif ou de la technologie).

- La qualité d'efficacité qui couvre le caractère subjectif de la QoE. Il apparaît en effet qu'une technologie peut être très performante en termes techniques mais cela peut ne pas être assez efficace pour satisfaire l'utilisateur ou répondre à ses attentes.
- Les attentes qui permettent de mesurer la qualité d'efficacité de manière adéquate. En effet, ce n'est que lorsqu'on a un aperçu des attentes de l'utilisateur qu'on peut savoir si une technologie fonctionne bien ou est assez suffisante pour cet utilisateur. Le niveau de satisfaction de ces attentes détermine alors la qualité d'efficacité.
- Le contexte dans lequel se trouve l'utilisateur influence directement ses attentes. Cinq types de contexte sont proposés: environnemental, personnel/ social, culturel, technologique et organisationnel.

Le modèle proposé a été construit avec l'intention de couvrir non seulement ce que le service fait, mais aussi ce que les personnes peuvent faire avec le service, ce qu'ils veulent/pensent à faire avec le service et ce qu'ils attendent du service, le contexte dans lequel ils ont l'intention de l'utiliser, et jusqu'à quel point il répond à leurs attentes.

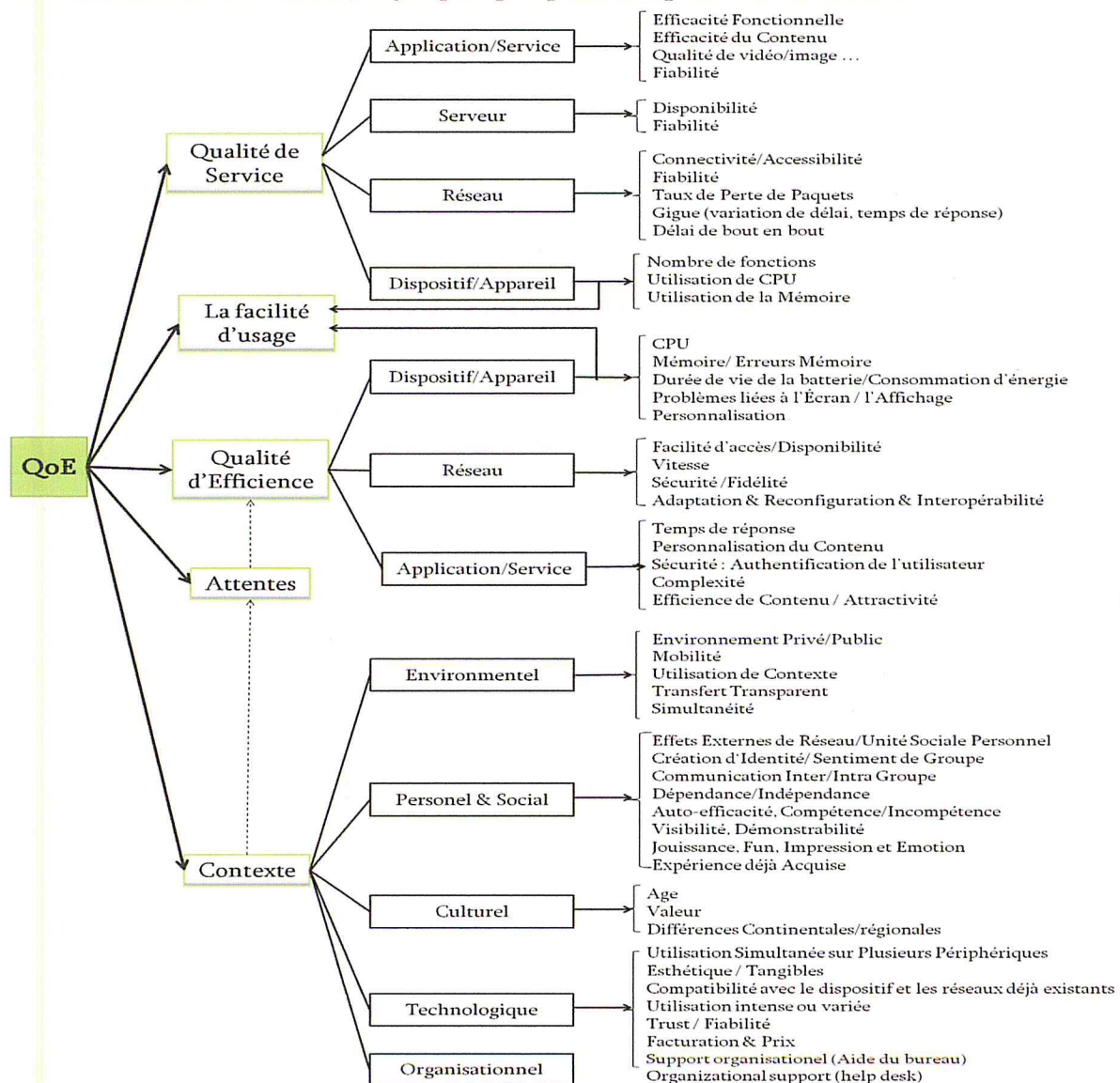


Figure 5: Concept multidimensionnel de la QoE (13).

I.3.2.4 Le Modèle de Diallo (14)

Selon Diallo, il existe beaucoup de facteurs qui influent sur la QoE des services de vidéos. Ces facteurs dépendent de la qualité de la vidéo à la source, de livraison de contenu sur le réseau (perte de paquets, gigue, délai, bande passante), du contexte utilisateur qui comprend des informations sur l'utilisateur (préférence utilisateur, emplacement), des caractéristiques du terminal (taille de l'écran, capacité de CPU) et enfin de la qualité vidéo perçue à travers le terminal du client. Nous avons essayé d'illustrer ces facteurs dans la figure 7.

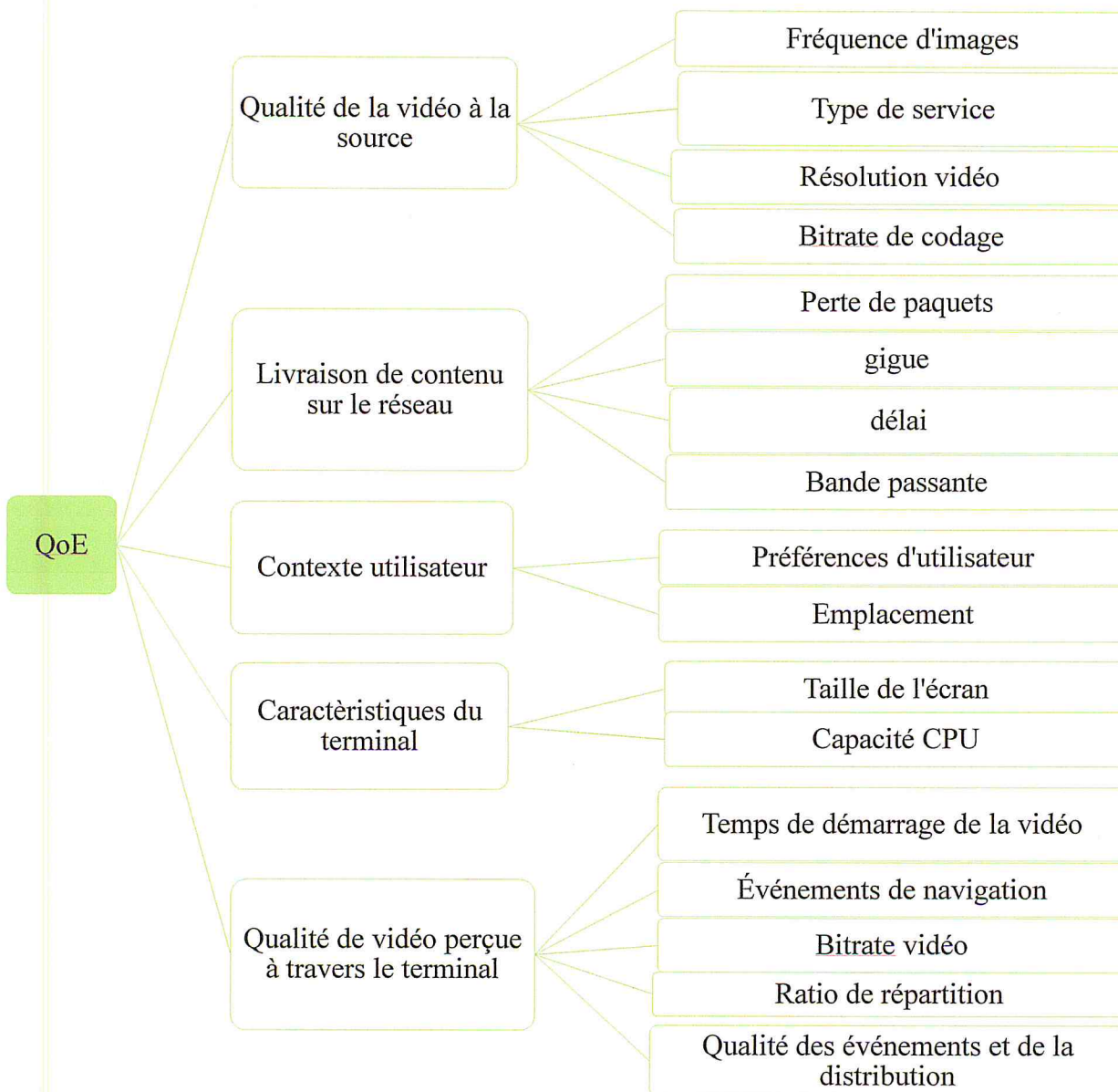


Figure 6: Les facteurs influant la QoE dans le modèle de Diallo.

I.3.2.5 Le modèle de Chen et al (15)

Dans le contexte de la transmission vidéo, Chen et al. trouvent que la QoE couvre les facteurs de bout en bout (c.-à-d. De l'hôte vers les clients) qui influencent l'expérience de l'utilisateur sur l'application ou le service vidéo comme le montre la figure 8. En général, ces Facteurs d'Influence ou (IF) peuvent être résumés en trois catégories :

- les FIs du système de communication qui sont liés au contenu, au média, au réseau et au périphérique),
- les FIs contextuelles qui incluent le contexte de la tâche, le contexte social ainsi que le contexte technique et d'information.),
- les FIs humains qui sont des composantes essentielles qui définissent la QoE comme une expérience individuelle de quelqu'un.

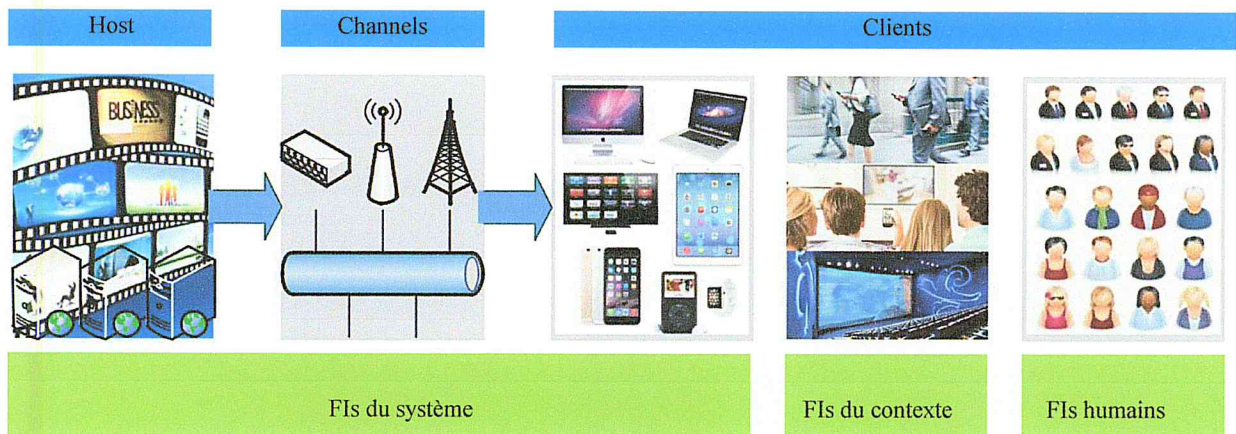


Figure 7: Facteurs d'Influence du Chen et al (15).

I.3.2.6 Le modèle de Musthaq (16)

La QoE est très subjective par nature, en raison de sa relation avec le point de vue de l'utilisateur et son propre concept de «bonne qualité». La capacité de mesurer QoE donnerait aux opérateurs de réseaux un certain sens de la contribution de la performance du réseau à la satisfaction globale du client en termes de fiabilité, de disponibilité, d'évolutivité, de rapidité, de précision et d'efficacité. D'après (16) les facteurs sont classés en quatre types comme suit.

- Les paramètres du réseau : les paramètres de la QoS reflètent l'état du réseau et son impact direct sur la QoE. La dégradation de chaque paramètre réseau conduit à la dégradation de la qualité du streaming vidéo. Parmi ces paramètres, on cite : la perte des paquets, qui est causée par la congestion du réseau ou l'arrivée tardive des paquets, la bande passante (la quantité de données numériques transmises par unité de temps), Latence (délai de bout-en-bout) définit le temps que parcourt un paquet entre l'émission et la réception et la gigue (la variation du délai dans le temps d'arrivé des paquets à destination),
- Les caractéristiques de contenu : les propriétés et les caractéristiques du contenu vidéo ont un impact considérable sur la QoE. Parmi ces paramètres, la disponibilité de la vidéo comme est considéré comme étant le premier facteur influençant la qualité perçue du service, la résolution d'image (le rapport entre le nombre de pixels de l'image et la taille réelle de sa représentation sur un support physique), le frame rate (Image par seconde), la complexité du mouvement, le contenu et la livraison de la vidéo ainsi que le codec,

- Les caractéristiques de la machine (terminal) : avec les progrès de l'industrie des télécommunications, des services multimédias en streaming sont déployés sur différents types de terminaux: Smartphone, Ordinateur portable, Tablette, PDA et TV. Ces équipements se différencient les uns des autres par leur résolution, leur processeur et leur taille d'affichage, parmi les paramètres de l'appareil affectant la QoE nous avons l'écran d'affichage, la charge du CPU et le logiciel de visualisation
- Les facteurs psychologiques : Cependant, la QoS ne détermine pas un élément important de la perception humaine sur le service actuel, car le comportement de l'être humain est difficile à prévoir. Il devient nécessaire pour les fournisseurs de services réseau de prendre en compte un grand nombre de paramètres et de métriques qui reflètent directement le comportement émotionnel de l'utilisateur. Dans le cas d'un service multimédia spécifique, la perception humaine varie d'une personne à l'autre. L'estimation du niveau de qualité dépend de nombreux facteurs, qui sont liés aux préférences et à l'environnement de cette personne. Certains de ces facteurs sont classés comme suit: les caractéristiques de l'utilisateur (âge, sexe, connaissance du fond, langue, familiarité avec tâche). Les caractéristiques de la situation et les conditions d'affichage (espace bruyant, nombre simultané des utilisateurs, à la maison, dans une voiture), et le comportement de l'utilisateur et son attention sur la vidéo en cours de lecture. Nous avons essayé de montrer les principaux facteurs du modèle de Mushtaq dans la figure9.

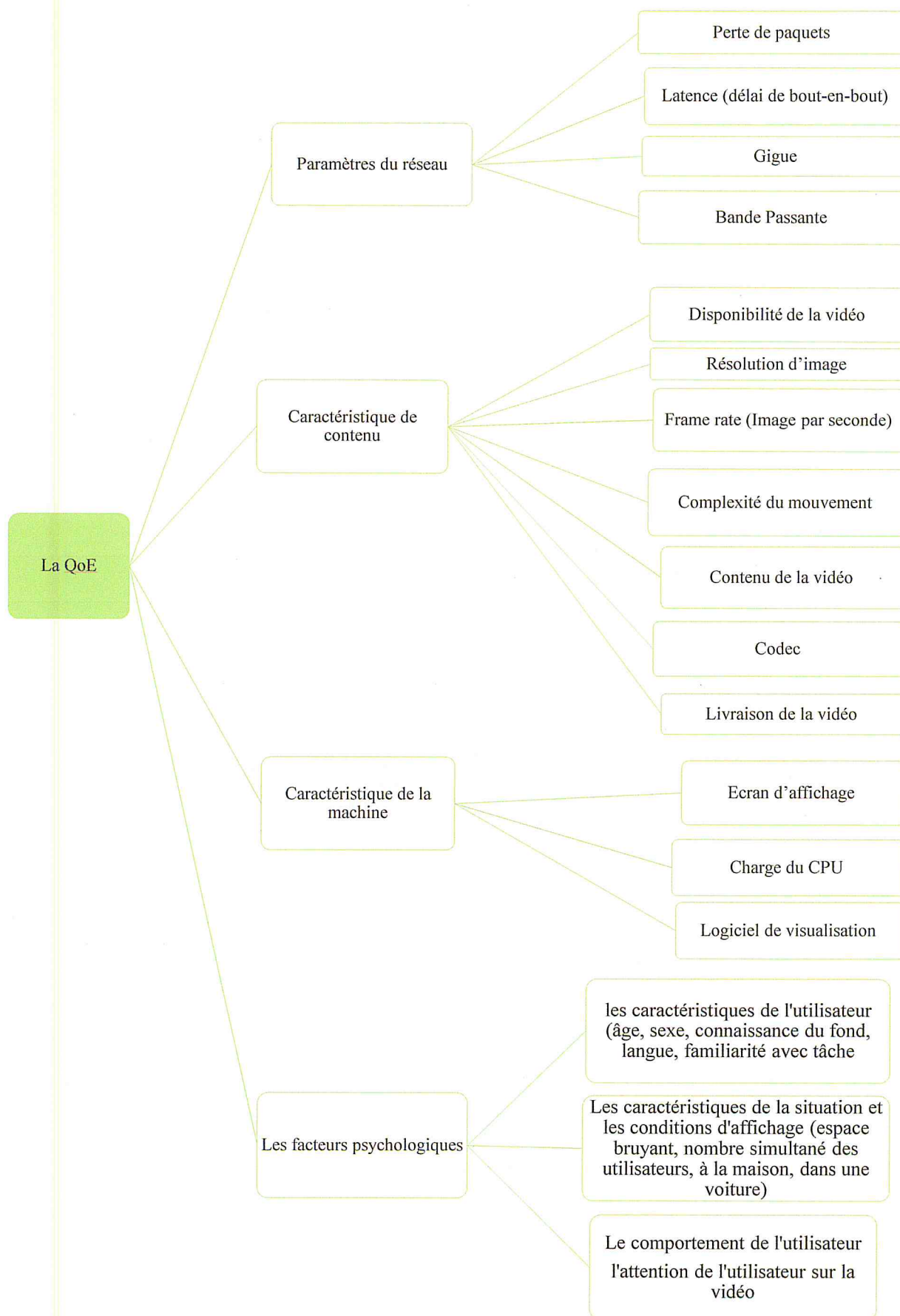


Figure 8: Le modèle de métrique de Musthaq.

I.3.2.7 Modèle d'Amour et al. (17)

Selon (17), Amour et al. ont proposés une nouvelle architecture hiérarchique et modulaire pour classer les facteurs d'Influence (IF) de la QoE. Cette architecture considère un grand nombre de FI qui peut être étendu selon les attentes futures des acteurs du système. D'une part, leur classification se compose de trois couches : Entité de surveillance, Entité de surveillance contextuelle et Entité de surveillance de l'expérience. D'autre part, chaque couche est composée de modules : La couche d'Entité de Surveillance contient quatre modules: paramètres des terminaux, paramètres d'infrastructure, paramètres QoS et paramètres des applications. La couche d'Entité de surveillance contextuelle couvre toutes les informations contextuelles sur l'utilisateur, elle se compose de quatre modules: Paramètres d'emplacement, paramètres de mobilité, paramètres réseau et paramètre psychologiques. La troisième couche Entité de surveillance de l'expérimentation montre la méthodologie expérimentale et la collecte de commentaires. La figure10 si dessous montre l'architecture de ces différentes couches.

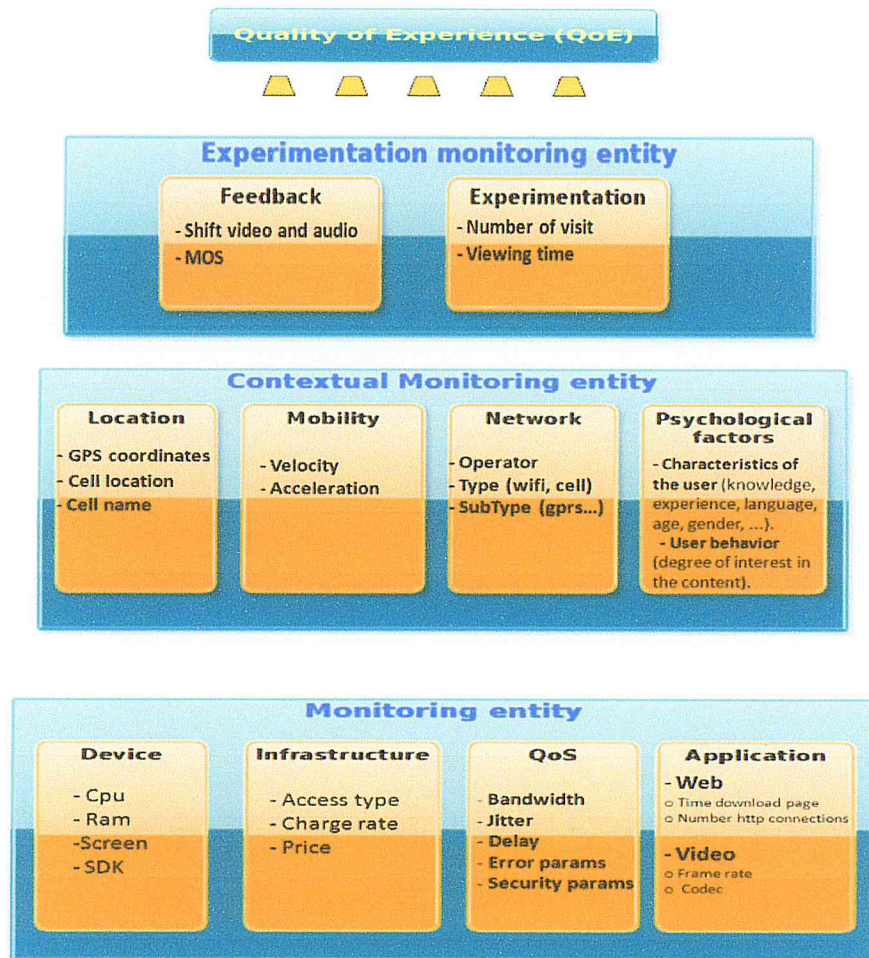


Figure 9: Le modèle de métrique d'Amour (18).

I.3.3 Synthèse

Après avoir passé en revue les principaux modèles de métriques de la QoE, nous constatons qu'il existe plusieurs facteurs (ou paramètres) peuvent influencer et/ou altérer la QoE.

Nous pouvons classer les principaux facteurs en six catégories :

- a. les caractéristiques de la vidéo source (e.g. largeur et hauteur de la trame, image rate,..)
- b. les paramètres de l'application codec (codage/décodage) de la vidéo (e.g. bitrate de codage,...),
- c. les paramètres de performance du réseau (e.g. bande passante, délai, gigue, taux de perte,),
- d. les paramètres de service (e.g. disponibilité, fiabilité, sécurité,),
- e. les paramètres de terminal (e.g. écran, cpu,),
- f. le profil de l'utilisateur (age, sexe, expérience, contexte,).

Selon ces facteurs, nous faisons une comparaison entre les modèles cités précédemment comme il montre le tableau suivant :

Modele/ catégorie	a	b	c	d	e	f	Autres
IUT_T (11)	-Débit binaire -Résolution d'image	-MPEG - CBR -VBR	-Délai -Gigue -Bande passante	-Disponibilité	-	-Emotion -Expérience	-Facturation de service
DSL forum (12)	-La couleur -Débit binaire -Résolution d'image	-Standard (MPEG-2, MPEG4 AVC, VC-1) -CBR/VBR	-Délai -Taux de perte- -Gigue	-Fidélité de l'information -Facilité d'utilisations -Instantanéité -La sécurité -Fiabilité -Disponibilité	-	-	-
Marez & Moor (13)	-	-	-Taux de Perte, Délai, Gigue, Connectivité-	-Disponibilité, Fiabilité Temps de réponse, Personnalisation Du contenu, Accessibilité	-Nombre de Fonction, Utilisation de cpu, Utilisation de la mémoire, Ecran / affichage	-	-
Diallo (14)	-Fréquence d'images. -Résolution vidéo -Type de	-	-Perte de paquets -Gigue -Bande passante -délai	-	-Taille de l'écran -Capacité CPU	-Préférence -utilisateur -Emplacement	-

	service -Ratio de répartition -Qualité des éléments.						
Chen et al (15)	-Exigence temporelle -Profondeur de couleur 2D, 3D -Temps de lecture	- H.265 encodeur	-Bande passante -Délai -Gigue -Perte -Taux d'erreur -Débit	-Fiabilité du contenu	-	-Emplacement -Mouvement	-
Musthaq (16)	-Contenu de la vidéo -Résolution d'image -Frame -Complexité mouvement	-MPEG -H.262, -H.263, -H.264, -WVVD, -WMV3,	-Perte de paquets -Délai -Gigue -Bande passante	-Disponibilité	Ecran d'affichage, Charge de cpu	-Facteurs Psycho- logiques	-
Amour (17)	-Taux d'image -Décalage audio/vidéo	-84 Kbps et 1 Mbps avec AV C=H:264	-Bande passante -Délai -Gigue -Taux d'erreur	-	-Cpu -Ram -Ecran -SDK	-Emplacement (lieu géographique, emplacement GPS) -Comportement	-

Tableau 2: Tableau comparatif entre les modèles de métriques.

Les sept modèles de métriques exposés ci-dessous ont été proposés dans le but d'identifier les principaux facteurs qui influents sur la QoE des services de streaming vidéo.

Nous remarquons que pour les caractéristiques de la vidéo source, la plupart des modèles mettent l'action sur la résolution d'image et le débit binaire. Alors que pour les paramètres de performance de réseau, Nous trouvons que tous les modèles ont pris en considération ces paramètres, notamment, la bande passante, le délai, la gigue et la perte. D'autre part, en comparant les paramètres de service, nous trouvons que les modèles s'intéressent à la disponibilité du service ainsi que sa fiabilité. Pour les paramètres de terminal (écran d'affichage, capacité du cpu, ..) et les caractéristiques de l'utilisateur (âge, sexe,..), nous constatons qu'ils ne sont pas vraiment prisent en considération dans tous les modèles.

En conclusion, après avoir fait cette étude comparative, *nous avons su définir la première version de notre modèle de métriques qui contient les principales catégories d'FIIs de la QoE et qui sont (a) les caractéristiques de la vidéo source, (c) les paramètres de performance du réseau, (e) les paramètres du terminal et (f) le profil de l'utilisateur.*

1.3.4 Méthodes de mesure :

Une bonne méthode d'évaluation de la QoE peut aider à contrôler la qualité des services vidéo et le renforcement de l'expérience de ses utilisateurs. En raison de son rôle fondamental, un grand nombre de mesures objectives de la QoE ont été proposées au fil du temps. A la différence des évaluations objectives, qui fonctionnent d'une façon automatique, on trouve l'évaluation subjective qui nécessite une intervention humaine. Dans qui suivent nous détaillerons ces deux méthodes de mesures.

1.3.4.1 Méthode Subjective de mesure (9)

Les méthodes subjectives consistent principalement en des expériences de laboratoire, dans lesquelles un certain nombre de personnes sont exposées à des échantillons de vidéo dégradée qu'ils doivent ensuite noter. Une telle note doit refléter leur perception, ou opinion, de la qualité de la vidéo. Après un filtrage statistique, la moyenne des opinions (Mean Opinion Score, MOS) est ensuite considérée comme la mesure de la qualité de l'échantillon (Tableau 3). Ces méthodes ont été normalisées dans les recommandations P.910 de l'ITU-T et BT.500 de l'ITU-R

5	4	3	2	1
Excellent	Bien	Acceptable	Pauvre	Mauvais

Tableau 3 : Mean Opinion Score (MOS)

Pour la vidéo, les méthodes subjectives semblent les plus naturelles. Cependant, elles sont coûteuses en termes de temps et de moyens humains les rendant ainsi difficiles à reconstituer aussi souvent qu'on le souhaite. En effet, elles nécessitent au moins une population de 15 personnes ayant une bonne acuité visuelle pour chaque série de contenus vidéo évalués. Les durées des tests dépendent du nombre d'échantillons à évaluer, de leurs durées et de la nature

de la tâche assignée aux participants. Ainsi, les méthodes subjectives ne permettent pas de surveiller en temps réel les applications vidéo. De telles difficultés, associées aux méthodes subjectives, ont donné lieu au développement des méthodes objectives qui, bien que moins performantes, sont beaucoup plus pratiques et moins coûteuses.

1.3.4.2 Méthodes de mesure Objectives (9)

Les méthodes objectives font référence à des formules et des algorithmes établis dans le but d'automatiser le processus d'évaluation. Elles se basent sur la mesure d'une ou plusieurs métriques objectives de la QoE. La validation de ces méthodes se fait par une comparaison de ses notes à celles recueillies lors des tests subjectifs. Cette comparaison est principalement établie à l'aide du coefficient de corrélation entre les notes objective et les notes subjectives, appelées MOS (Mean Opinion Score). Ce coefficient de corrélation doit être bien choisi afin de présenter de façon la plus réaliste possible la relation entre les métriques objectives et la perception de l'utilisateur.

Plusieurs travaux ont été menés dans ce sens, donnant ainsi naissance à une pléthore de modèles d'évaluation de la QoE. Ces modèles, connus aussi sous l'appellation des modèles de prédiction, sont présentés dans la section suivante.

1.4 Modèles de prédiction

Dans la littérature, plusieurs modèles de prédiction QoE ont été proposées (19). La majorité de ces modèles utilisent les techniques ML afin de trouver la relation (ou corrélation) entre la QoE et les paramètres de performance du réseau ou les paramètres de QoS. Dans (19), les auteurs proposent une classification des modèles de corrélation QoE-QoS en se basant sur le type d'apprentissage comme illustré dans la figure 11.

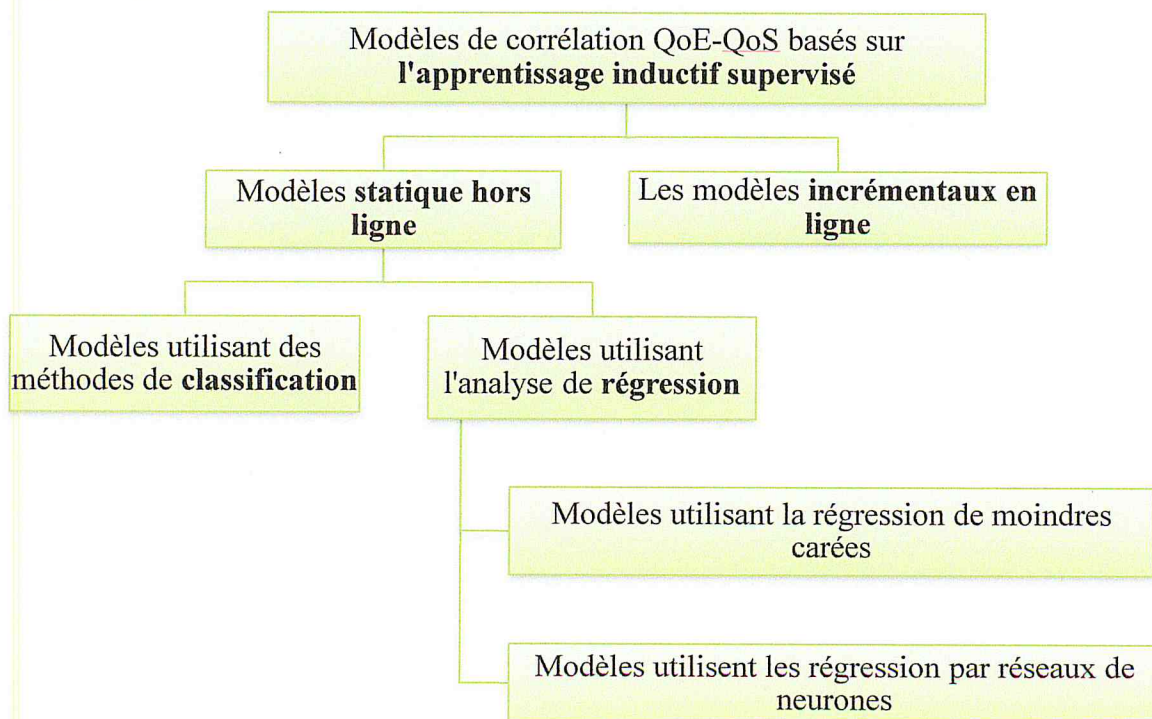


Figure 10: Catégorisation des modèles de corrélation QoE-QoS (19).

De cette catégorisation, nous trouvons plusieurs formes des modèles de prédiction: linéaire, non linéaire (logarithmique, puissance, exponentielle, réseaux de neurones, arbres de décision, forêts d'arbre, réseaux bayésiens,...). Pour plus de détails, nous invitons le lecteur à consulter la référence (19). Par ailleurs, la validation de ces différents modèles se fait en général par expérimentation dans un laboratoire où les chercheurs mettent en place leur plateforme d'expérimentation qui permet de contrôler un ou plusieurs FIs afin d'étudier leur impact sur la QoE. Cette phase d'expérimentation est coûteuse en termes du temps, de main d'œuvre et autres. C'est pourquoi, notre objectif est de faciliter la tâche de chercheur en lui offrant une plateforme générique lui permettant de collecter le maximum des FIs afin de réaliser ses études.

I.4 Conclusion

Dans ce premier chapitre, nous avons exposé les principaux axes de notre travail en commençant par les services de streaming ensuite la QoE. Dans la première partie, nous avons présenté l'architecture et les protocoles de service VoD, le champ de notre travail. Ensuite, dans la deuxième partie, nous avons passé en revue plusieurs définitions et modèles de métriques de la QoE. Nous avons retenu de cela, la définition de la QoE donnée dans le cadre du séminaire de Dagstuhl et les principales catégories des facteurs d'Influence de la QoE qui sont : les caractéristiques de la vidéo source, les paramètres de performance du réseau, les paramètres de terminal et enfin le profil de l'utilisateur.

Dans ce domaine de recherche, le but est d'arriver à évaluer cette QoE de manière automatique en étudiant la relation entre la QoE et un ou plusieurs de ses facteurs d'Influence. On parle d'ici des modèles de prédiction qui nécessitent en général une série d'expérimentation pour être validé. Pour cela, chaque chercheur conçoit sa propre plateforme d'expérimentation qui lui permet de réaliser ses tests en collectant les principaux facteurs d'influent à étudier. Dans le chapitre suivant, nous allons passer en revue les principales plateformes d'expérimentations.

Chapitre II : Travaux Antérieurs

II.1 Introduction

Dans la littérature, plusieurs travaux de recherche ont été menés afin d'étudier l'impact d'un ou de plusieurs facteurs sur la QoE. Pour ce faire, il a toujours été nécessaire de réaliser des expériences dans un environnement de test contrôlé ou réel pour construire une base (ou un échantillon) de données contenant les valeurs des facteurs influant la QoE ainsi que l'évaluation de la QoE par des sujets humains. Pour réaliser ces expériences, on a besoin d'une plateforme qu'on appelle « plateforme d'expérimentation de QoE ». Par la suite, nous définissons cette plateforme, ensuite nous décrivons les principales plateformes existantes dans la littérature.

II.2 Définition

Une plateforme en informatique est une base de travail à partir de laquelle on peut écrire, lire et développer et utiliser un ensemble de logiciels. Elle peut être composée :

- Du matériel : x86, x86-64, SPARC, PowerPCn IA-64, etc.
- Du système d'exploitation : Linux, Unix Windows, Mac OS X , ou un simple noyau parfois temps réel , etc.
- D'outils logiciel de développement : bibliothèque logicielle, API voire Framework : Struts, Hibernate, etc. débogueurs, éditeur de texte, compilateurs et éditeur de liens voire Environnement de développement intégré : Visual Studio, Eclipse, etc.
- De gestion de projet : gestionnaire de bugs, de version : Apache Subversion, etc.
- De SGBD : MySQL, PostgreSQL , Oracle, etc.
- D'un serveur web : Apache, IIS, etc.
- D'un serveur d'application : JOnAS, Tomcat, JBoss, etc.
- D'environnement d'exécution de logiciel : le système d'exploitation et son interpréteur de commandes, dans le cas de phase de compilation ou machine virtuelle ou interpréteur.

Les plateformes informatiques sont généralement conçues, développées, construites, mises en service et maintenues par des constructeurs informatiques ou des prestataires de services (20).

Dans ce contexte, nous pouvons définir une plateforme d'expérimentation de QoE comme étant *une plateforme informatique qui permet aux chercheurs de mener des expériences, en environnement contrôlé et/ou réel, dans le but d'étudier l'impact d'un ou de plusieurs facteurs sur la QoE. Cette plateforme permet ainsi de générer une base (ou un échantillon) de données contenant les valeurs des facteurs influant la QoE ainsi que l'évaluation de la QoE par des sujets humains. Elle permet aussi aux chercheurs d'analyser les résultats récoltés afin de définir et/ou valider leurs modèles de prédiction de la QoE.*

II.3 Environnement de test

L'environnement de test dans lequel se fait les tests ou expériences, devrait être impartial, ne devrait pas être affecté par un bruit extérieur et devrait être proche du scénario réel. La

plupart des travaux de recherche réalisent leurs expériences dans un des laboratoires, on parle ici de *l'environnement contrôlé*. Néanmoins, *l'environnement réel* (connu sous l'appellation anglaise « crowdsourcing ») devient populaire aussi (21). Dans cette section, nous présentons ces deux types d'environnement.

II.3.1 Environnement contrôlé (22 ; 23)

L'expérience de laboratoire fournit un environnement contrôlé pour effectuer des tests subjectifs pour évaluer la qualité multimédia. Les différents paramètres associés au test, comme le niveau de bruit, la distance entre l'écran et les utilisateurs, la taille de l'écran, etc., peuvent être facilement contrôlés selon les exigences. Cependant, les expériences faites dans le laboratoire ont des limites telles que

- un coût élevé en termes de temps et de main-d'œuvre
- la diversité limitée des participants.

Une expérience contrôlée nécessite des semaines pour préparer des tests, recruter des utilisateurs, planifier des intervalles de temps pour superviser les expériences, etc. Les utilisateurs doivent également être physiquement présents dans le laboratoire pour effectuer le test. Généralement, les tests de laboratoire sont effectués en université ou dans un laboratoire de recherche, de sorte que les participants au test sont les étudiants ou les chercheurs. Cela limite la diversité des participants: la plupart d'entre eux sont des garçons qui sont jeunes et instruits, etc.

II.3.2 Environnement réel (23)

Dans l'environnement réel ou crowdsourcing, les expériences peuvent être effectuées à distance et il y a peu de contrôle sur l'environnement du participant. C'est une méthode assistée par le logiciel qui est généralement réalisée via une application Web. Cette méthodologie consiste principalement à collecter une évaluation subjective de la qualité grâce à une diffusion omniprésente via Internet. Les différents tests d'écran et les mécanismes de détection de triche sont utilisés dans une expérience de crowdsourcing pour rendre le test fiable et résistant. Les expériences de Crowdsourcing ont attiré l'attention pour remplacer les besoins des tests faits dans les laboratoires et ces expériences offrent une corrélation prometteuse avec l'ancien. Cela permet à un chercheur d'obtenir des opinions d'une grande variété de sujets. Dans (24) les auteurs spécifient quelques étapes à prendre en considération dans le cas de l'environnement réel :

- Utiliser le logiciel Collin qui ne nécessite pas d'installations administratives.
- Utiliser des instructions de langage simple qui attirent les locuteurs non natifs pour sous-estimer les instructions sans utiliser de ressources de langage additif (dictionnaires).
- Spécifier une bonne durée pour les expériences car les participants à la crowdsourcing sont beaucoup moins engagés que les participants labos et ils peuvent se retirer des travaux à tout moment.
- Inclure des sessions de formation appropriées.
- Intégrer un canal de rétroaction pour contacter l'expérimentateur
- Utiliser l'enregistrement des événements.
- Inclure des contrôles de fiabilité dans la conception de test.

- Inclure des contrôles de fiabilité pendant le test.
- Inclure des contrôles de fiabilité après le test.

II.3.3 Comparaison:

Nous présentons une comparaison entre ces deux types d'environnement dans le tableau suivant :

Contrainte de comparaison	Environnement contrôlé	Environnement réel
les Facteurs d'Influence (FI) la QoE	<ul style="list-style-type: none"> - Un nombre moyen des FI à collecter - Les FI sont bien à contrôler et surveiller - Les FI sont faciles à mesurer 	<ul style="list-style-type: none"> - Un nombre élevé des FI à collecter - Certains FI sont difficiles, voire impossible à contrôler et à surveiller - Certains FI sont difficiles à mesurer.
Coût d'installation	<ul style="list-style-type: none"> - Quasiment gratuit du fait que le serveur peut être une machine comme celles des utilisateurs et donc pas de coût d'hébergement des données. - Coût élevé en termes de temps et de main-d'œuvre (préparer des tests, recruter des utilisateurs, planifier les tests dans le laboratoire) 	<ul style="list-style-type: none"> - Coût relatif à l'hébergement de la base de données contenant les informations des clients - Coût réduit en termes du temps et de main d'œuvres (inviter les utilisateurs)
Nombre des participants et la taille de l'échantillon	<ul style="list-style-type: none"> - Il y a peu de participants dans une zone géographique restreinte et petite. - Taille de l'échantillon d'ordre des centaines 	<ul style="list-style-type: none"> - Une grande variété des participants - Taille de l'échantillon d'ordre milles

Tableau 4: L'environnement contrôlé vs l'environnement réel.

II.4 Architecture générique de la plateforme d'expérimentation

En général, l'architecture d'une plateforme d'expérimentation de QoE est composée de trois parties (Figure 12):

- « Utilisateurs » où on trouve les participants aux tests (ou testeurs) avec leurs terminaux.
- « réseaux de communication » qui peut être un réseau local contrôlé dans le cas d'un environnement contrôlé ou un réseau ouvert comme Internet dans le cas d'un environnement réel.

Cette plateforme est composée d'un réseau local privé avec 3 ordinateurs portables connectés à un routeur sans fil. L'un des ordinateurs portables a été utilisé comme un serveur du multimédia (multimédia server) et l'autre comme un client (receveur) qui reçoit les flux multimédia. Le troisième ordinateur a été utilisé en tant qu'une passerelle.

La vidéo a été projetée sur un écran plat LCD via la sortie VGA de l'ordinateur portable du client. Le téléviseur est monté à l'aide d'un support mural à une hauteur de 3,5 mètres du sol. Les téléspectateurs de la vidéo se tenaient à une distance de 6 pieds de l'écran ayant un angle de vision de 70 degrés à 110 degrés. Le lecteur multimédia open source VLC Player a été utilisé pour la diffusion de la vidéo et pour lire le contenu à la réception. Les deux ordinateurs portables sont équipés d'un système d'exploitation Windows tandis que l'ordinateur passerelle est équipé d'Ubuntu. Ubuntu a été utilisé pour simuler la variation des paramètres réseaux en utilisant «netem» qui est intégré dans de nombreuses distributions Linux.

Au total, deux clips vidéo ont été utilisés, un clip vidéo avec des mouvements rapide d'un match de football et une autre avec des mouvements plutôt lent. Le taux de trame vidéo est de 30 images par seconde, la durée d'un teste a été fixé à 12 secondes. Le contenu vidéo a été codé avec le / MPEG-4 norme de codage vidéo H.264 et en streaming en utilisant le protocole UDP sur le réseau sans fil IEEE 802.11n.

L'expérience a été réalisé avec 24 sujets; parmi eux 6 étaient des femmes et 18 étaient sujets de sexe masculin âgés de 20 à 35 ans. Les sujets ont été fournis avec un questionnaire et ils ont été invités à fournir leurs informations de profil et leur avis après teste sur la qualité de la vidéo. La qualité vidéo est mesurée avec une échelle de 5 points d'intervalle des étiquettes à chaque extrémité comme par exemple la note 1 (Pire / Fortement insatisfait) à 5 (excellent / Fortement satisfaits).

II.5.2 Plateforme du BENYETTOU (27)

Benyettou (27) a travaillé avec une architecture similaire de Laghari. Le dispositif expérimental est constitué d'un serveur de streaming vidéo, d'un routeur wifi et de trois clients vidéo. Trois écrans d'affichage ont été sélectionnés pour lire la vidéo du côté client. Un émulateur réseau (*NETEM*) fait varier les paramètres réseau au niveau des 3 interfaces réseau du serveur. La figure 14 représente le schéma global de la plateforme:

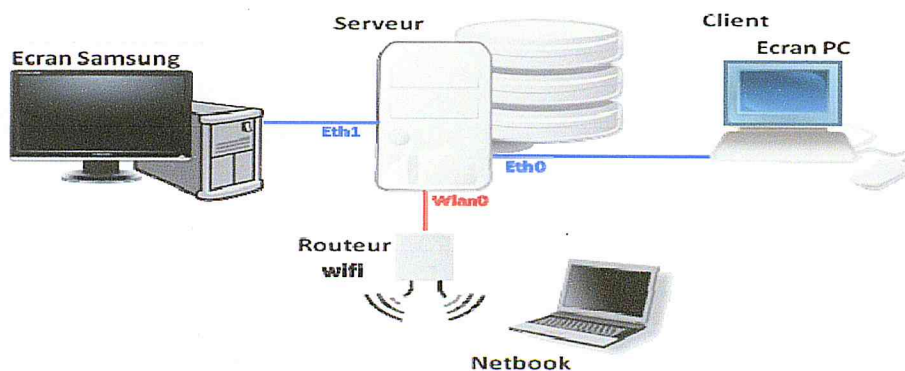


Figure 3: Plateforme de Benyettou (27).

- « Serveurs » où on trouve l'ensemble des fonctionnalités offertes par la plateforme comme streaming vidéo, gestion de bases de données, évaluation de QoE, gestion d'administration etc.

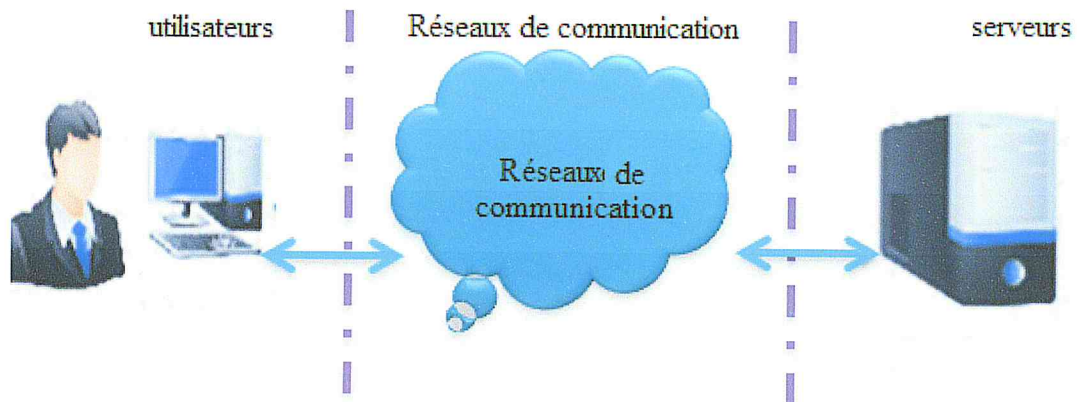


Figure 1 : Architecture globale de la plateforme d'expérimentation.

II.5 Principales plateformes existantes

Dans la littérature, plusieurs plateformes ont été mises en œuvre afin d'étudier l'impact d'un ou de plusieurs facteurs sur la QoE et de proposer (ou valider) un (le) modèle de perte de QoE. Dans ce qui suit, nous allons décrire les principales plateformes.

II.5.1 Plateforme de Laghari (environnement contrôlé) (25)

Pour étudier l'effet combiné de paramètres de QoS (la perte de paquets, la gigue, le délai et le débit binaire vidéo) et de type du contenu (football et conteneur vidéo) sur la QoE dans les réseaux sans fil, Lagharai (25) a mené une étude subjective basée sur les recommandations de l'UIT-T (26), ainsi que la méthodologie d'évaluation qualitative sur la plateforme illustrée dans la figure 13 suivante.

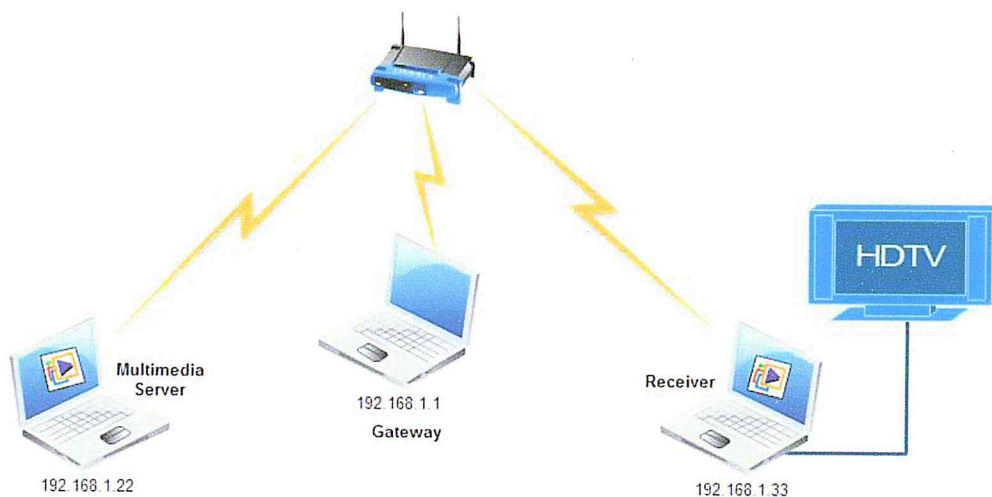


Figure 2: Architecture de l'expérience (25).

L'expérience s'est faite sur trois vidéos choisies avec précaution : la première vidéo avec peu de mouvement, la deuxième avec mouvement moyen et la troisième avec mouvement rapide. Les séquences ont été découpées du même fichier source afin qu'elles disposent des mêmes propriétés comme la résolution le taux d'image par seconde et le type d'encodage.

Les paramètres affectant la QoE qui ont été choisies sont : les paramètres du codec, les paramètres du réseau et le type de contenu. Les paramètres du codec sont le taux d'images par seconde (FR : Frame Rate), et le débit de flux sortant de l'encodeur (SBR : bitrate sender). Les paramètres réseau choisis ont été la Bande passante (LWB) le taux de perte (PER : Packet error rate) et la gigue (jitter), De plus, un paramètre sur la complexité du mouvement des séquences est pris en compte : Peu de Mouvement, Mouvement Moyen et Mouvement Rapide. Les séquences vidéo ainsi que les paramètres choisis sont récapitulé dans le tableau 5 suivant :

Ecran	Séquences	SBR(kb ps)	FR(fp s)	LWB(mbps)	PER(%)	Jitter(ms)
Ecran Samsung	Vidéo1(P.M)	600,80	10,15,	1,1.5,2	0,5,10	0,5,10
Ecran de PC	Vidéo2(M.M	0,1000	30			
Netbook) Vidéo3(M.R)					

Tableau 5 : Combinaisons d'ensemble de données (27).

L'auteur a invité 29 personnes pour réaliser un test subjectif. Tous les spectateurs étaient des étudiants de différentes disciplines âgées de 17 à 40 ans avec peu ou aucune expérience de ce type de tests. Chaque utilisateur a testé les 81 échantillons en utilisant la méthode ACR pour évaluer la qualité.

Les tests subjectifs sont composés de deux phases: la phase initiale (les instructions et la phase de formation) et des sessions de tests. Une session de formation sur l'application a été incluse avant le test réel. Le but de cette session était d'expliquer pleinement ce qui est requis par les utilisateurs pour les familiariser à ce type de test.

II.5.3 Plateforme de Laghari (environnement réel) (25)

Cette plateforme conçue aussi par Laghari dans (25) se déroule dans un environnement réel comme Internet, C'est une première dans le domaine des plateformes d'expérimentation de QoE pour les services multimédia. C'est une tentative de fournir un outil d'évaluation QoE de l'industrie et de la recherche, le projet est encore à ses débuts et est open source et sujet à des modifications et des améliorations constantes. La plateforme nommée QoM (QoE pour les services Multimédia), est conçue en modèle J2EE. Ses principales composantes sont un administrateur, un serveur de streaming vidéo, une interface web pour les clients, une interface pour l'administrateur et une base de données QoE comme montré dans la figure 15. Les capteurs de paquets ou sniffer sont déployé sur les deux côté du service c'est-à-dire côté serveur et coté client et un serveur VLC assure un service VOD aux clients à travers le navigateur web.

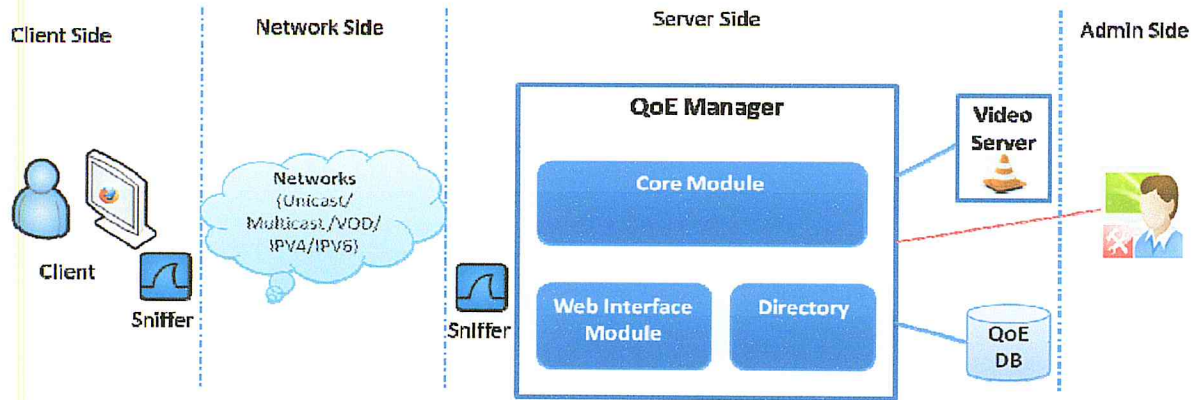


Figure 4: Plateforme QoM (25).

L'interface client Web (figure 16) est conçu pour faciliter aux utilisateurs de regarder des vidéos en ligne et de donner leurs notes QoE. Elle se compose d'informations du profil utilisateur (âge, sexe, profession, pays, et l'emplacement), de section « note QoE » (échelle quantitative de 5 points et des commentaires qualitatifs) et de section « vidéo ». La section « vidéo » permet de regarder n'importe quel contenu vidéo donné. La machine côté client devrait avoir la dernière version de Firefox ou Google Chrome avec VLC plug-in en utilitaire. Comme l'interface client est basée sur navigateurs Web, il peut fonctionner sur n'importe quel système d'exploitation qui prend en charge Firefox et Google Chrome. De plus, la machine cliente doit contenir un utilitaire Wireshark afin que le client puisse capturer de données concernant le trafic réseau. Chaque fois, un utilisateur souhaite regarder la vidéo, il / elle sera alloué une session aléatoire avec un nombre unique. Il doit insérer l'adresse IP du client avant de pouvoir exécuter un test vidéo.

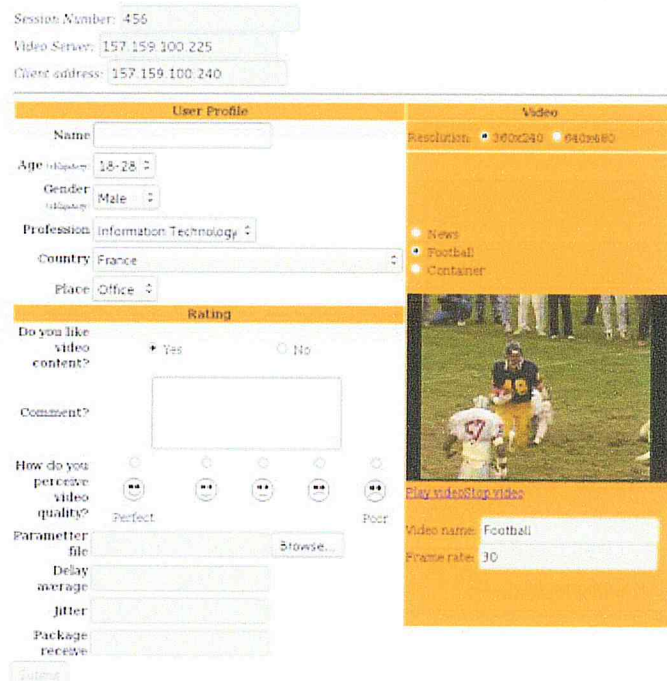


Figure 5: L'interface graphique de la plateforme QoM (25).

II.5.4 Plateforme de Mushtaq et al. (16)

Le laboratoire vivant est un nouveau concept utilisé dans différents recherche en mettant l'accent sur l'expérience de l'utilisateur. Il essaie d'apporter le laboratoire aux bénévoles dans son contexte réaliste. Mushtaq. et al. (16) avaient comme but de leur travail d'évaluer la QoE dans un environnement réel (crowdsourcing) en collectant un grand ensemble de données. Pour cela, la plateforme est déployée sur Internet comme étant réseau de communication, ce qui offre l'opportunité de la participation à distance des utilisateurs. Pour ce faire ils ont proposé un outil qui peut être utilisé efficacement pour le test de crowdsourcing afin d'étudier la QoE des utilisateurs en temps réel. Cet outil travaille en deux parties. La première partie est de détecter la vidéo sur le site, et après la fin de la vidéo, il faut que l'utilisateur commente la qualité perçue du streaming vidéo. La deuxième partie utilise la méthode objective pour mesurer et stocker les facteurs d'Influence de la QoE (voir le modèle de Mushtaq, chapitre I, section I.2.2.6) en temps réel L'outil enregistre le degré de satisfaction des utilisateurs, il est testé en utilisant le portail Web YouTube.

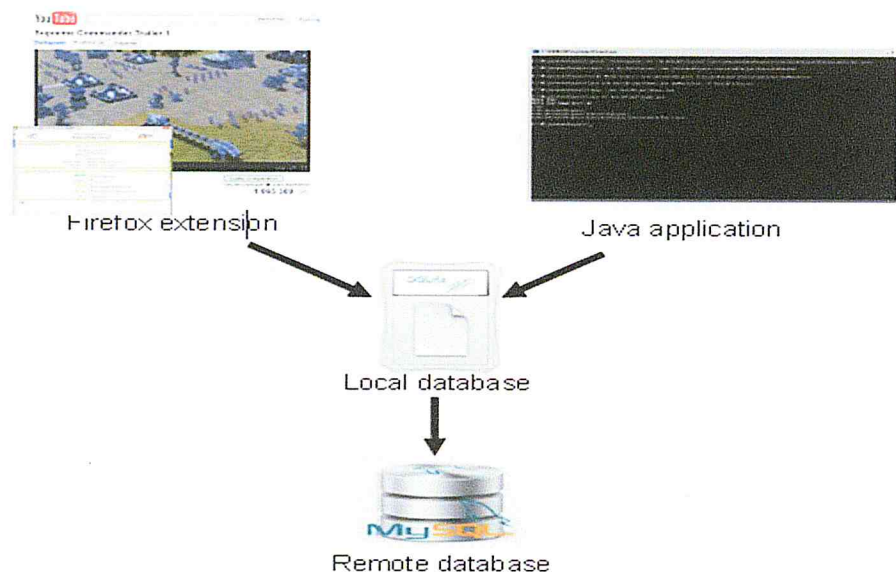


Figure 6: L'architecture de la plateforme de Musthaq et al(16).

La figure 17 montre l'architecture de cette plateforme. Elle est basée sur deux modules majeurs: extension Firefox et application Java. L'extension FireFox est développée en version javascript qui utilise principalement le développement Web. Il représente un complément de langage XML dans une extension Firefox pour améliorer et enrichir l'interface graphique de l'application. Cette extension permet d'ajouter un bouton au bas (libelé Quality of Experience) de la vidéo en ligne, comme le montre la figure 18.

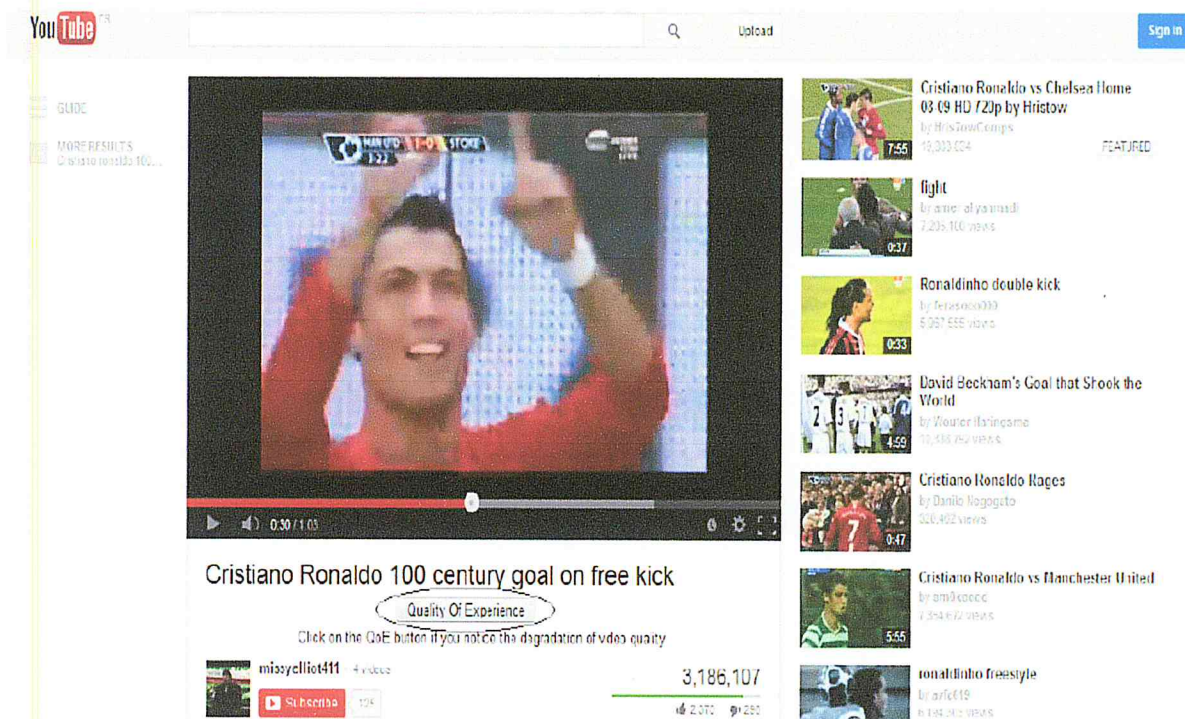


Figure 7: Framework Implémentation (16).

The image shows a web form titled 'Quality of Experience'. At the top, it features logos for UPEC (Université Paris-Est Créteil) and LISSI (Laboratoire Informatique des Systèmes de Télécommunications). The form includes several input fields: 'Name (*)', 'Age (*)', 'Profession (*)', 'Sex (*)' (with radio buttons for Female and Male), 'Video viewing frequency (*)' (with radio buttons for Rarely, Every week, and Every day), and 'Video content is (*)' (with radio buttons for Interesting and Not interesting). Below these fields is a star rating section with five options: 'Excellent, Perfect video without problems' (5 stars), 'Good, Video with minor problems' (4 stars), 'Fair, Video with some problems that affected the quality' (3 stars), 'Poor, Video with several problems that affected really the quality' (2 stars), and 'Bad, Very bad video quality' (1 star). A 'Submit' button is located at the bottom left of the form.

Figure 8: Formulaire de commentaire de l'utilisateur (16).

Lorsque l'utilisateur clique sur le bouton, le formulaire de retour (figure 19) s'ouvrira, afin de recueillir les commentaires de l'utilisateur, de stocker les informations initialement dans une base de données locale et ensuite de transférer ces informations vers un serveur distant.

À la fin du test de crowdsourcing, lorsque tous les paramètres sont extraits des deux modules (application Java et extension Firefox), les données collectées passent de la machine utilisateur à un serveur éloigné pour enquêter sur la QoE de l'utilisateur.

II.5.5 Plateforme d'Amour et al (17)

Pour étudier l'effet combiné de paramètres de QoS (la perte de paquets, la gigue, le délai et le débit binaire vidéo) et de type du contenu (football et conteneur vidéo) sur la QoE , AMOUR (17) a développé une plateforme nommée CLIF (framework Lissi-lab contrôlé) qui est une plateforme open source pour aider les chercheurs à construire une grande base de données QoE pour les services de streaming vidéo (Youtube). La figure 20 montre l'architecture de cette plateforme.

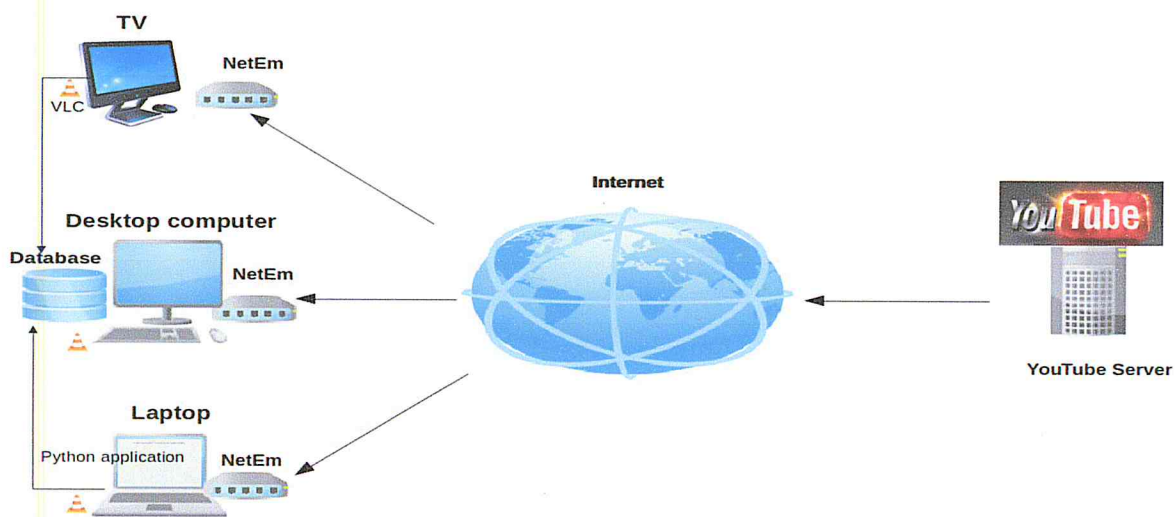


Figure 9: Plateforme CLIF (17).

Pour considérer toutes les IF de QoE qui peuvent avoir un impact sur la perception de l'utilisateur (voir le modèle d'Amour, chapitre I, section I.2.2.7), ils ont utilisé les logiciels suivant: VideoLAN Client player (VLC), Netem (Network Emulator), Python, MySQL serveur et User Feedback (UF).

Plusieurs types de vidéos ont été utilisés (sport, film, documentaire, nouvelles, musique) dans les expériences (figure 21)

	News	Sports	Documentary	Music	Cartoons	Lessons	Movies	Games
Very eventful								
Medium								
Low eventful								

Figure 10: Capture d'écran de vidéos utilisées (17).

De plus, pour évaluer l'impact de la QoS réseau (retard, perte et taux), la plate-forme utilise l'émulateur de réseau NetEm. Ces trois facteurs de QoS prennent les valeurs indiquées dans le tableau 6 et sont variés d'une manière aléatoire.

QoE IFs	Valeur
delay(ms)	0, 100, 200, 400
rate(kbitsrs)	256, 512, 768, 1024, 1536
loss(%)	0, 5, 10

Tableau 6 : Variation de QoS (17).

La figure 22 présente des captures d'écran sur les interfaces de la plateforme.

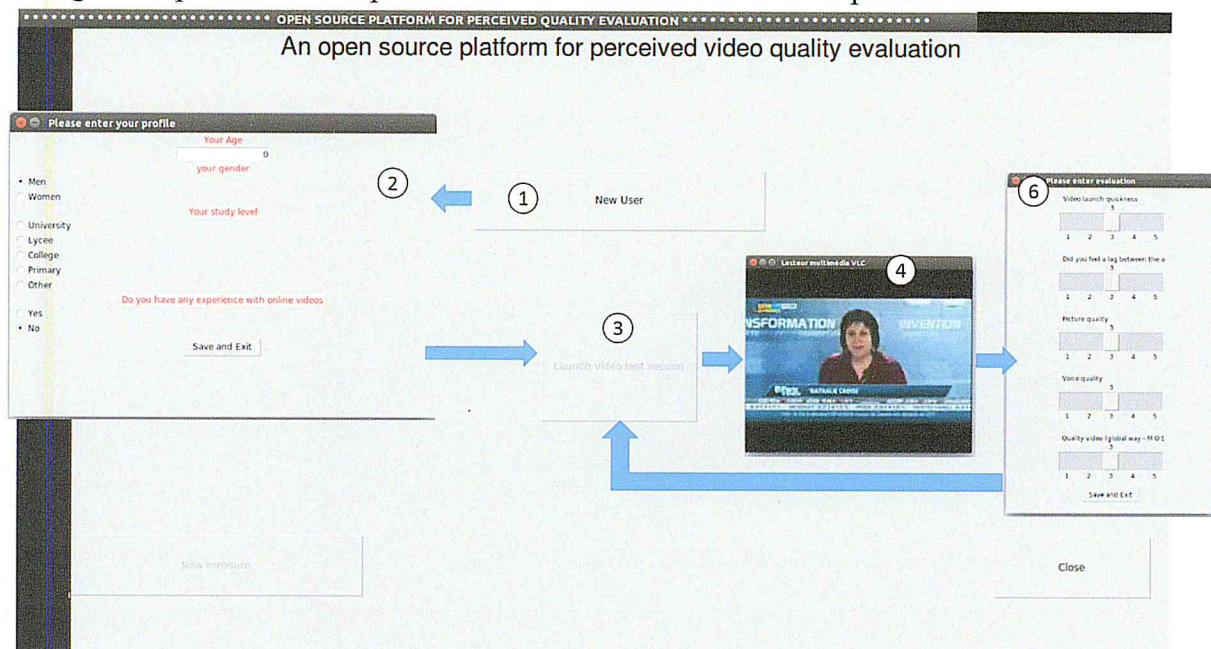


Figure 11 : Des captures d'écran sur les interfaces de la plateforme (17).

II.5.6 Discussion

Après avoir vu les travaux qui existent, voici un tableau comparatif entre ces plateformes décrites ci-dessous (tableau 7).

Plateforme	Type d'environnement	Les facteurs d'influence IF	Modèle de Mesure de la QoE	Modèle de Prédiction	Nombre des participants/ Taille de l'échantillon	Open source	L'architecture		
							Partie utilisateur	Partie réseau	Partie Serveur
Khalil Laghari (26)	Contrôlé	-Paramètre de QoS (la partie de paquet, la gigue, le délai et la variation du délai et le débit binaire -Le type de contenu	La méthode ACR	Algorithme d'apprentissage basé sur la régression linéaire multivariante	24 Individus	-	écran plat LCD, lecteur vidéo VLC, Linux	Réseau local +émulateur (NETEM)	Serveur streaming (VLC), serveur BDD
Lahouari benyettou (28)	Contrôlé	-Les paramètres du Codec, les paramètres du réseau (la bande passante, le taux de perte et la gigue) , Le type de contenu	La méthode ACR	- se base sur un modèle d'apprentissage basé sur les systèmes adaptatifs neuronal d'inférence floue et l'analyse de régression non linéaire	29 individus	-	3 écrans d'affichage (netbook, ecran pc , ecran samsung)	Réseau local + Emulateur réseau (NETEM)	Serveur streaming
Khalil Laghari (26)	Réel	Tous les paramètres QoS : le délai,	La méthode ACR	Algorithme d'apprentissage	-	Oui mais le	Interface web	Réseau	Serveur BDD,

		packet loss, packet reoder, bit rate,		e basé sur la régression linéaire multivariante		code source n'est pas disponible	(firefox) + sniffer Wireshark	Internet	serveur Streaming (VLC), sniffer Wireshark
Musthaq (17)	Réel	-Les paramètres de QoS - Le Type de contenu	La méthode ACR	-	-	-	Interface (TV, Desktop, laptop),	Emulateur réseau	Service BDD, service Streaming (youtube)
Amour (18)	Réel	-paramètre QoS du réseau -Caractéristiques des terminaux	La méthode ACR	Il compare plusieurs méthode dont : NB, DT,RF, SVM,RNN, ANFIS, Average.	81 couples.	Oui et le code source est disponible	Interface web, application Python	Réseau ouvert+ émulateur réseau netem	Serveur BDD, serveur Streaming (Youtube)

Tableau 7 : Tableau comparatif entre les plateformes

Les cinq plateformes exposées ci-dessous ont été développées dans le but d'étudier l'impact de certains facteurs sur la QoE des services de streaming vidéo. Nous constatons que :

- la majorité (3/5) des plateformes ont été développées dans un environnement réel.
- pour les Facteurs Influent, toutes les plateformes faites leur étude sur les paramètres réseaux (la gigue, la bande passante, le taux de perte et le délai) par contre en trouve la majorité s'intéressent au type de contenu (3/5 plateformes) et seulement une plateforme prend en considération les caractéristiques des terminaux.
- Toutes les plateformes utilisent la méthode ACR pour mesurer la QoE.
- Toutes les architectures des plateformes se composent de trois parties :
 - o la partie utilisateurs qui se diffère au niveau de type d'interface graphique utilisée : web (à base de navigateur mozilla) ou application java.
 - o la partie réseaux où la plupart des plateformes ont utilisé l'émulateur réseau NetEm.
 - o la partie serveurs qui se compose dans la majorité des plateformes d'un serveur de BDD et d'un serveur streaming VLC.
- Les plateformes ne sont pas toute open source, nous avons pu récupérer le code source d'une seule plateforme (celle d'Amour)¹ où nous avons trouvé des difficultés pour l'installer et la déployer.

En conclusion, si un chercheur veut faire une nouvelle étude sur l'impact d'un ou plusieurs FIs sur la QoE dans le cadre d'un type de service donné, il doit implémenter et mettre en place sa propre plateforme pour collecter son échantillon et cela en raison du manque des plateformes libres et gratuites. Pour cela, notre travail consiste à mettre en œuvre une plateforme générique et extensible permettant de faciliter la tâche des chercheurs. Dans cette plateforme, nous allons essayer de combiner les avantages des cinq plateformes, à savoir :

- Capacité de fonctionner en environnement local et réel. A notre connaissance, aucune plateforme ne permet de faire cela. Laghari propose une architecture pour chaque environnement, nous allons essayer de proposer une seule architecture pour les deux types d'environnement ;
- Collecte du maximum des FIs, entre autres les paramètres de QoS et le type de contenu ;
- Utilisation de la méthode ACR pour mesurer la QoE ;
- Utilisation des logiciels collins (navigateur mozilla, vlc, netem) au niveau des utilisateurs ;
- Mise en disposition du code source et des tutoriaux d'installation et d'utilisation pour des futurs déploiements.

¹ Il faut contacter les auteurs pour avoir le code source

II.6 Conclusion

Dans ce chapitre, nous avons commencé par définir ce qu'est une plateforme d'expérimentation de QoE, ensuite nous avons expliqué les deux types d'environnement (contrôlé et local) dans lequel on déploie cette plateforme. Puis, nous avons évoqué les principales plateformes réalisées dans la littérature. Enfin, nous avons terminé avec une comparaison de cinq plateformes et une discussion qui nous a permis de dégager les principales caractéristiques de notre plateforme. Cette plateforme sera conçue dans le chapitre suivant.

Partie III

Etude Conceptuelle

« Après avoir cerné les différents principes du projet, l'étude conceptuelle est menée suivant les variantes et les axes décelés, permettant ainsi de recenser et de définir les scénarios possibles. »

Chapitre III : Etude Conceptuelle

III.1 Introduction

La réalisation de la plateforme doit être impérativement précédée d'une méthodologie d'analyse et de conception qui a pour objectif de permettre de formaliser les étapes préliminaires de son développement afin de rendre ce travail plus fidèle aux besoins de l'utilisateur. Une description générale des différentes fonctions de la plateforme s'impose en tout début de chapitre. Il sera question d'exposer les diverses fonctionnalités dans les deux environnements traités, local pour commencer et dans l'environnement réel.

Pour la conception de notre système, nous avons utilisé le cycle de vie en cascade en se basant sur le langage UML. Cependant la plate-forme n'est pas d'une grande complexité pour d'utiliser la totalité des diagrammes d'UML, nous n'utiliserons que les diagrammes que nous trouverons nécessaires. D'abord, nous avons traduit la liste des exigences en diagrammes de cas d'utilisation, qui permettent d'analyser et d'organiser les besoins, suivis par les diagrammes de séquence. Ensuite, nous avons étudié l'aspect statique avec les diagrammes de classes. Enfin, on expliquera dans le détail la phase la plus importante du travail qui est le processus de collecte d'informations ainsi que le processus d'exploitation des données.

III.2 Présentation de la démarche utilisée

Nous expliquons dans cette étape du chapitre, les besoins de notre plateforme afin de pouvoir passer à l'étape de conception et d'architecture. Nous présentons le cycle de vie que nous avons suivi pour la réalisation de ce projet. Nous illustrerons les solutions apportées par notre outil face aux problèmes posés, en se basant sur le langage UML (Unified Modeling Language) en utilisant le processus UP (Unified Process).

UP est une méthode de prise en charge du cycle de vie d'un logiciel développé en orienté objet. Il représente les étapes du cycle de vie sous formes de diagrammes UML.

III.2.1. Le cycle de vie

Le cycle de vie d'un logiciel est un ensemble séquentiel de phases, dont le nom et le nombre sont déterminés en fonction des besoins du projet, permettant généralement le développement d'un service ou d'un produit, en ce qui concerne notre projet nous avons suivi le modèle en cascade.

III.3 Modèle en cascade (28)

Le processus de développement que nous avons utilisé est le cycle de vie en cascade, ce modèle décrite par Royce en 1970 a été largement utiliser, pour la description générale des activités liées aux logiciels .il décrite le cycle de vie d'un logiciel par une suite de phase s'enchaîne dans un déroulement linéaire, depuis l'analyse des besoins jusqu'à la maintenance.

Le résultat de chaque étape est testé et on ne passe à l'étape suivante que lorsque l'étape actuelle est satisfaisante.

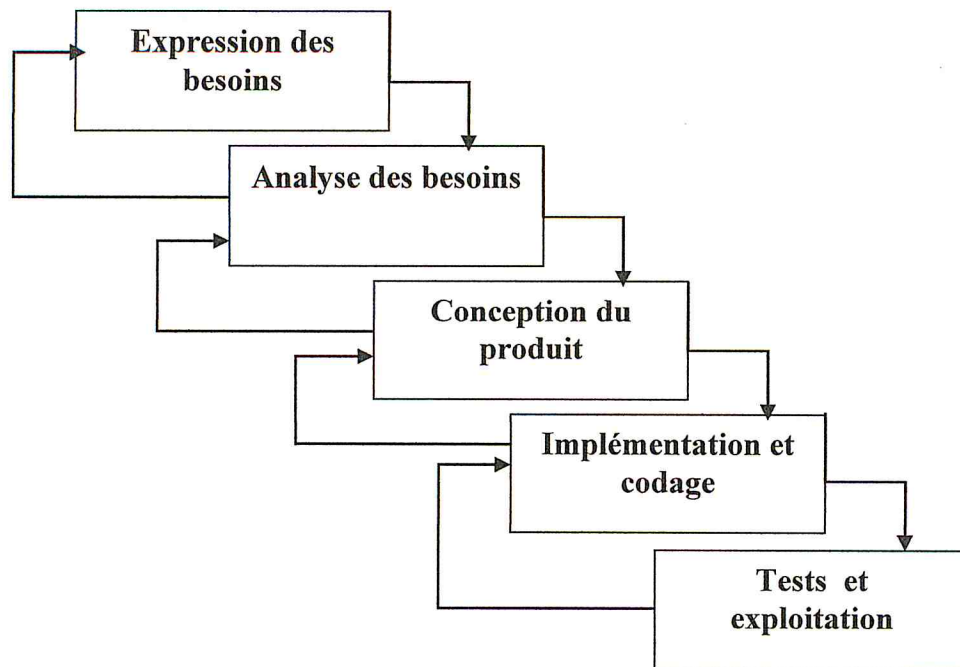


Figure 23 : Cycle de vie selon le modèle en cascade (28).

III.3.1 Expression des besoins

La spécification des besoins est une étape essentielle au début de processus de développement, elle consiste généralement à déterminer précisément les besoins des utilisateurs du système afin d'éviter de développer un logiciel non adéquat.

Cette étape ne préoccupe pas des solutions mais des questions : elle identifie le « quoi faire ? » Et identifie les entités de l'environnement du système. Pour modéliser ces besoins on utilise le diagramme des cas d'utilisation d'UML. (29)

III.3.2 Analyse

L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client, il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution.

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structures sous une forme qui facilite la compréhension (Scenario) en utilisant le diagramme de séquence d'UML pour représenter les interactions entre les objets. (30)

III.3.3 Conception

C'est la phase la plus importante du processus de développement d'un logiciel. Elle s'intéresse d'abord au « comment ? », à savoir la solution du problème énoncé.

La conception a pour but de décomposer le logiciel en module, de préciser les interfaces et les fonctions de chaque module. A l'issue de cette étape, on obtient une description de l'architecture du logiciel et un ensemble de spécifications de ces divers composants en utilisant les deux diagrammes de classe et d'activité d'UML. (30)

III.3.4 Implémentation

L'implémentation est le résultat de la conception pour implémenter le système sous forme de composants, c'est-à-dire, de code source, de scripts exécutables et d'autres éléments du même types.

Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous forme de code source. (31)

III.3.5 Tests

Les tests permettent de vérifier des résultats de l'implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun. (29)

III.4 Expression des besoins

Les différents besoins auxquels la phase réalisation doit prendre en considération peuvent se diviser en deux grandes parties à savoir, les besoins fonctionnels et les besoins non fonctionnels.

III.4.1 Besoins Fonctionnelles :

Il s'agit des fonctionnalités du système. Ce sont les besoins qui spécifient un comportement d'entrée / sortie du système. En effet, ces besoins répondent aux points précis du cahier de charge et prennent en considération surtout les besoins de l'utilisateur. Ce sont les "besoins primaires et primordiales" de la catégorie cible. Dans ce cadre, notre système doit répondre aux besoins fonctionnels suivants :

- Configurer une expérience en fixant sa durée, en choisissant la liste des vidéos à visualiser, et/ou en définissant les paramètres QoS du réseau.
- Exécuter un test de l'expérience en établissant une transmission entre un client et un serveur streaming via un réseau de communication. Ce réseau peut être :
 - Local où on parle d'environnement contrôlé qui consiste principalement en des expériences de laboratoire.
 - Etendu où on parle d'environnement réel dans lequel le serveur se trouve à distant et la communication se fait à travers d'Internet.
- Collecter les facteurs d'Influence de QoE suivants (voir la section I.2.3 du chapitre 1): le profil du testeur, les paramètres de QoS, les caractéristiques de vidéo, les caractéristiques du terminal et la note QoE de l'utilisateur.
- Enregistrer les informations collectées dans une base de données afin de l'exploiter ultérieurement.

III.4.2 Besoins non Fonctionnelles :

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes liés à l'implémentation (langage de programmation, de système d'Exploitation...)

ou à l'interopérabilité générale (ne pas bouffer toutes les ressources de la machine). En effet, ces besoins peuvent être fixés par le client (fonctions optionnelles), ou par le développeur. Elles ont été classées en deux catégories à savoir :

➤ **L'aspect sécurité :**

La sécurisation d'une plateforme web est aussi un équilibre délicat entre la facilité d'utilisation (ou utilisabilité) et la mise en œuvre de mesures de contrôle. Notre plateforme a l'avantage de combiner entre ces paramètres pour offrir à l'utilisateur tout le confort nécessaire lors de son utilisation, les mots de passe ne seront validés qu'après vérification de contenu c'est à dire chiffres et lettres ensemble et d'une certaine taille ce qui rendra la tâche difficile à un attaquant d'intercepter un mot de passe, la validation de l'inscription se fera par l'insertion d'un code visible à l'écran pour éviter les attaques à l'aide de robot d'essais.

➤ **L'aspect design :**

Le design est un aspect essentiel d'une plateforme web. Il véhicule d'emblée l'image du travail accompli par une société ou un particulier. Notre objectif est de créer pour l'utilisateur un thème graphique sur mesure qui d'une part facilite la navigation sur la plateforme, et d'autre part rend notre espace web attractif et ergonomique.

III.5 Diagrammes de cas d'utilisation :

Un diagramme de cas d'utilisation permet de décrire l'interaction entre l'acteur et le système. L'idée forte est de dire que l'utilisateur d'un système logiciel a un objectif quand il utilise le système ! Le cas d'utilisation est une description des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système. Les use case (cas d'utilisation) sont représentés par une ellipse sous-titrée par le nom du cas d'utilisation (éventuellement le nom est placé dans l'ellipse). Un acteur et un cas d'utilisation sont mis en relation par une association représentée par une ligne (32).

• **Identification des acteurs :**

UML n'emploie pas le terme d'utilisateur mais d'acteur. Un acteur représente un rôle joué par une entité externe (utilisation humaine, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données. Dans notre plateforme, nous pouvons distinguer les acteurs (rôles) suivants :

- **Testeur** : peut représenter un individu normal quel que soit son sexe mais de préférence d'une tranche d'âge un peu mure c'est-à-dire adulte de préférence entre 17 et 70 ans pour que la perception soit pertinente.
- **Administrateur** : est un technicien qui gère toutes les fonctions de la plateforme. Il possède des privilèges propres à son statut tel que la gestion des vidéos dans l'environnement contrôlé, la configuration des tests et la configuration du réseau local. Et il s'intéresse aussi aux résultats récoltés afin de les interpréter et de trouver des modèles de prédiction de la QoE.

Dans cette section, nous décrivons quatre diagrammes d'utilisation : un diagramme global et trois diagrammes détaillés. Le diagramme global (figure 23) englobe toutes les fonctionnalités qui peuvent être établit par l'utilisateur de la plateforme (tableau 8).

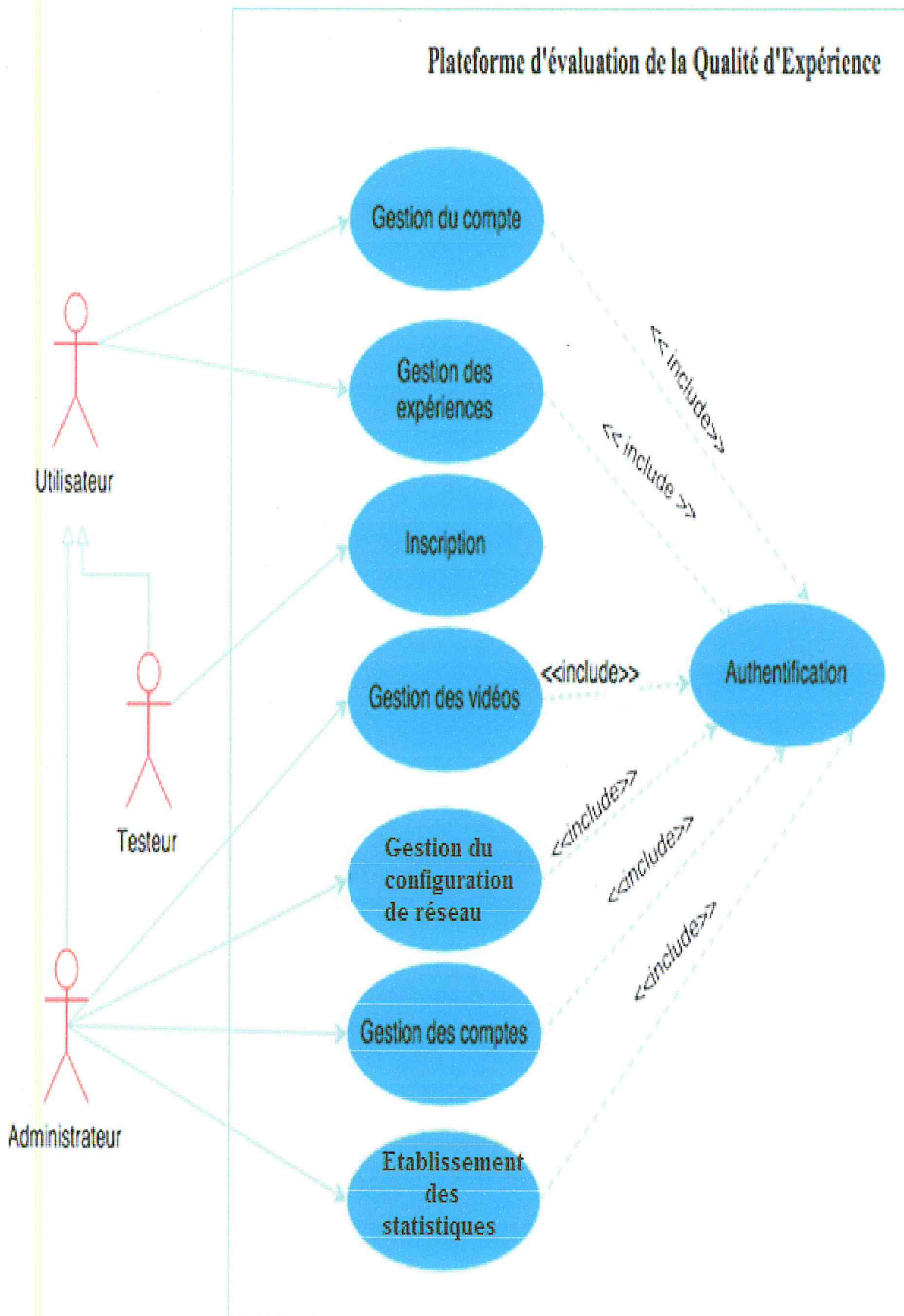


Figure 24: Description du diagramme Global de cas d'utilisation.

Cas d'utilisation	Acteurs	Description
Inscription	Testeur	chaque testeur a le droit de s'inscrire à la plateforme.
Gestion du compte	Utilisateur	Consulter et modifier les informations de son compte et se déconnecter.
Gestions des comptes	Administrateur	Consulter les profils des utilisateurs et activer/désactiver un compte.
Gestion de configuration de réseau	Administrateur	Créer et modifier un fichier de configuration réseau
Gestion des vidéos	Administrateur	Ajouter, consulter, modifier, supprimer une vidéo dans le serveur local.
Gestion des expériences	Administrateur Testeur	L'administrateur peut créer une nouvelle expérience, modifier ses paramètres, la supprimer, la consulter et la masquer de quelques utilisateurs. Le testeur exécute un test de l'expérience
Etablissement des statistiques	Administrateur	Consulter et exporter les résultats des expériences/tests
Authentification	Utilisateur	L'authentification est obligatoire pour pouvoir accéder à l'espace personnel de l'utilisateur.

Tableau 8: Description du diagramme global de cas d'utilisation.

Dans ce qui suit, nous allons détailler les fonctionnalités suivantes : gestion des expériences (figure 25 avec tableau 9), gestion du réseau local (figure 26 avec tableau 10) et Etablissement des statistiques (figure 27 tableau 11).

III.5.1 Gestion des expériences

Elle permet à l'administrateur de gérer les expériences et au testeur d'effectuer un test d'une expérience.

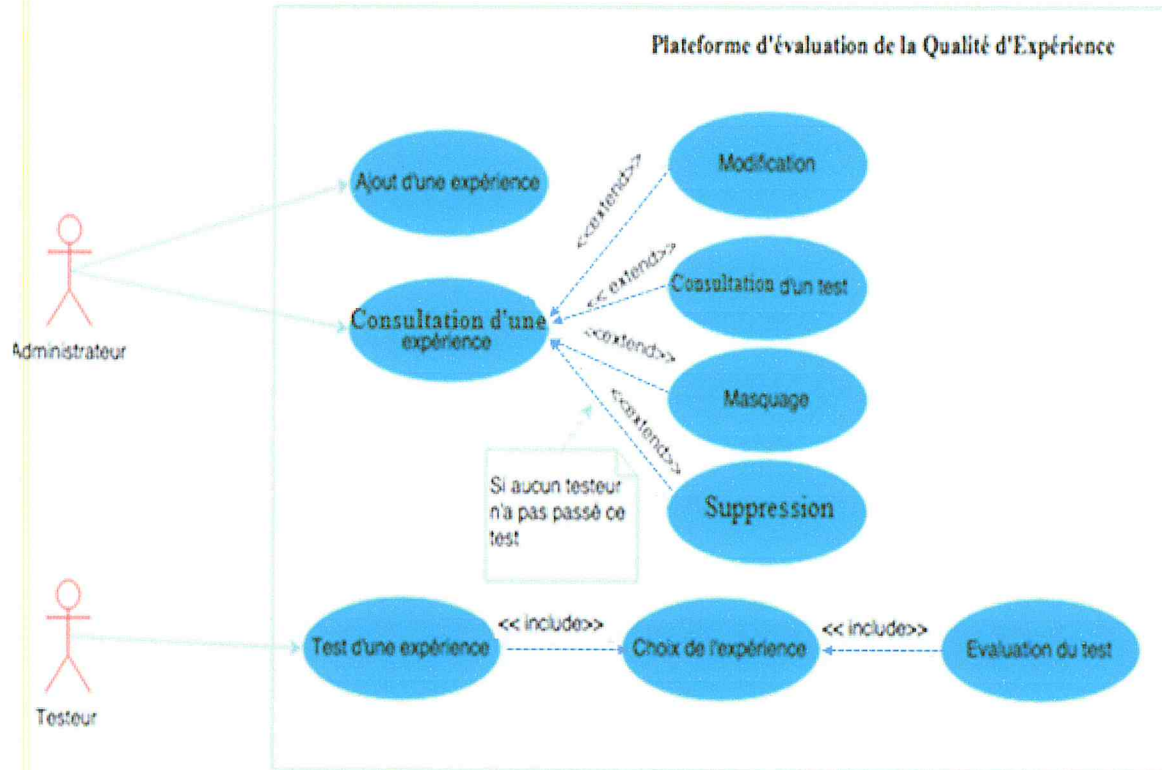


Figure 25 : Diagramme de cas d'utilisation « gestion des expériences ».

Cas d'utilisation	acteur	Description du cas d'utilisation
Ajout d'une expérience	Administrateur	Créer une nouvelle expérience en définissant sa durée, son fichier de configuration et la liste des vidéos à visionner et la liste des utilisateurs masqués.
Consultation d'une expérience	Administrateur	Consulter une expérience parmi les expériences qui existent pour : Modification Consultation d'un ou plusieurs tests Masquage Suppression
Masquage	Administrateur	Masquer une expérience. Le testeur ne peut plus effectuer un test d'une expérience masquée
Suppression	Administrateur	Supprimer une expérience si aucun testeur n'a pas passé un test de cette expérience
test d'une expérience	Testeur	Passer un ou plusieurs tests d'une expérience

Tableau 9: Description du cas d'utilisation " Gestion des expériences".

III.5.2 Gestion de configuration de réseau

Elle permet à l'administrateur de créer un fichier de configuration de réseau auquel les expériences y seront soumises ainsi que le modifier.

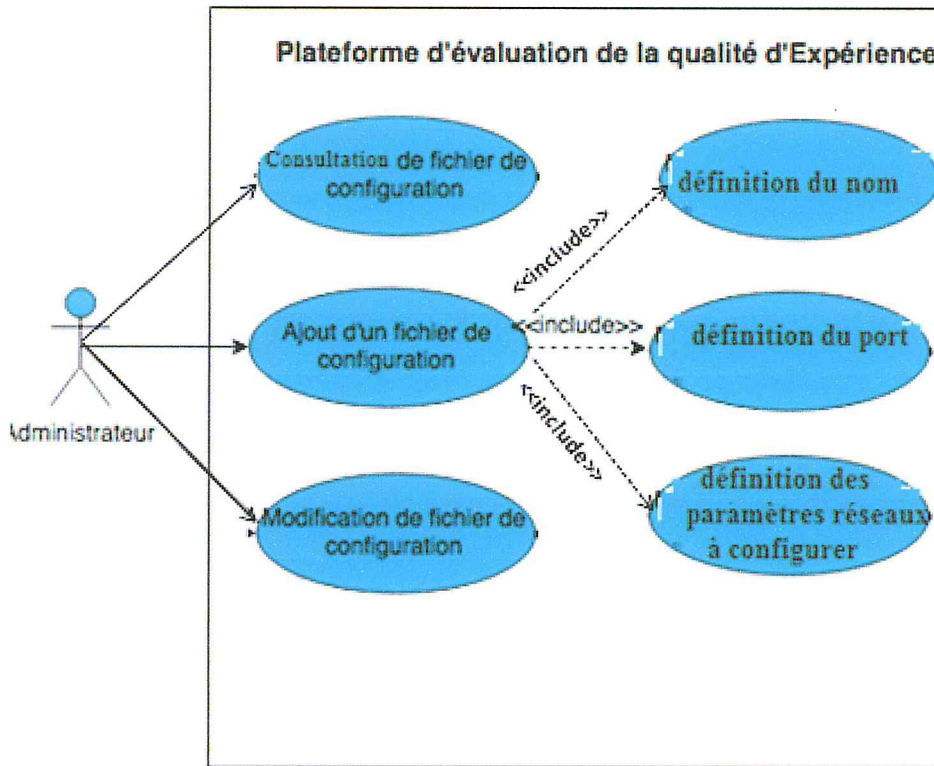


Figure 26 : Diagramme de cas d'utilisation " Gestion de configuration du réseau ".

Cas d'utilisation	acteur	Description du cas d'utilisation
Ajout d'un fichier de configuration	Administrateur	Ajouter un nouveau fichier de configuration des paramètres de réseau
Définition de nom	Administrateur	Lors de création d'un nouveau fichier, l'administrateur définit le nom du fichier.
Définition du port	Administrateur	Définir le port
Définition des paramètres réseaux à configurer	Administrateur	Choisir les paramètres de réseau pour les configurer.
Modification de fichier de configuration	Administrateur	Modifier le fichier de configuration
Consultation d'un fichier de configuration.	Administrateur	Modifier la valeur de la perte

Tableau 10: Description du cas d'utilisation " Gestion de configuration de réseau".

III.5.3 Etablissement des statistiques

Elle permet à l'administrateur d'établir les résultats obtenus après l'exécution d'un ou plusieurs tests.

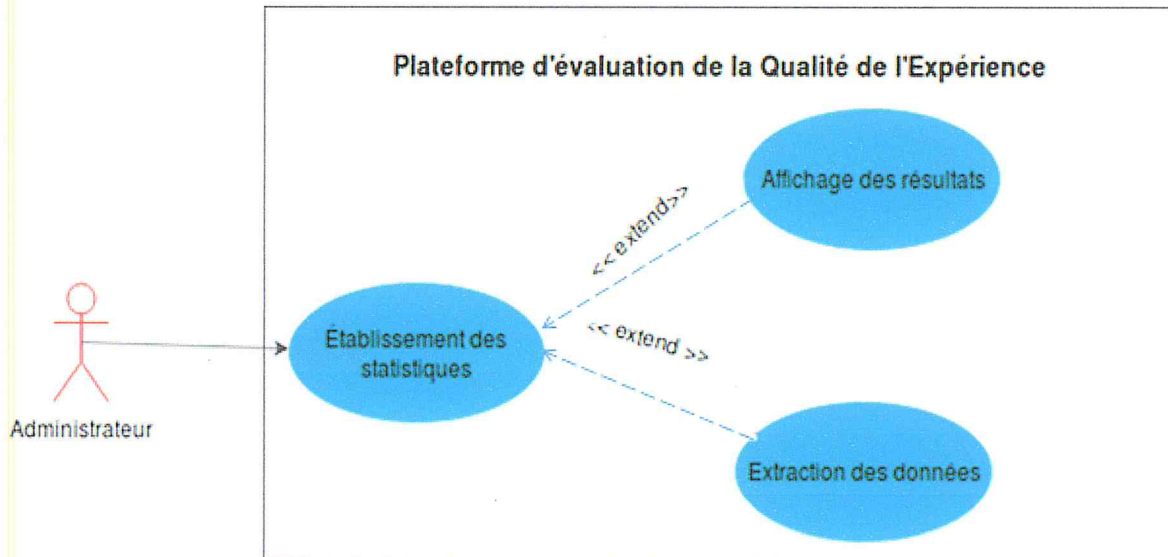


Figure 27: Diagramme de cas d'utilisation "Etablissement des statistiques".

Cas d'utilisation	acteur	Description du cas d'utilisation
Etablissement des statistiques	Administrateur	Etablir les statistiques
Affichage des résultats	Administrateur	Afficher selon des critères, les résultats sous format d'un tableau.
Extraction des données	Administrateur	Extraire les données dans un fichier Excel

Tableau 11: Description du cas d'utilisation « Statistiques ».

III.6 Diagrammes de séquence :

Le diagramme de séquences permet de cacher les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets. [33]. Par la suite, nous illustrons les diagrammes de séquence suivants : Ajout d'une expérience (figure 28), test d'une expérience (figure 29) et statistique (figure 30).

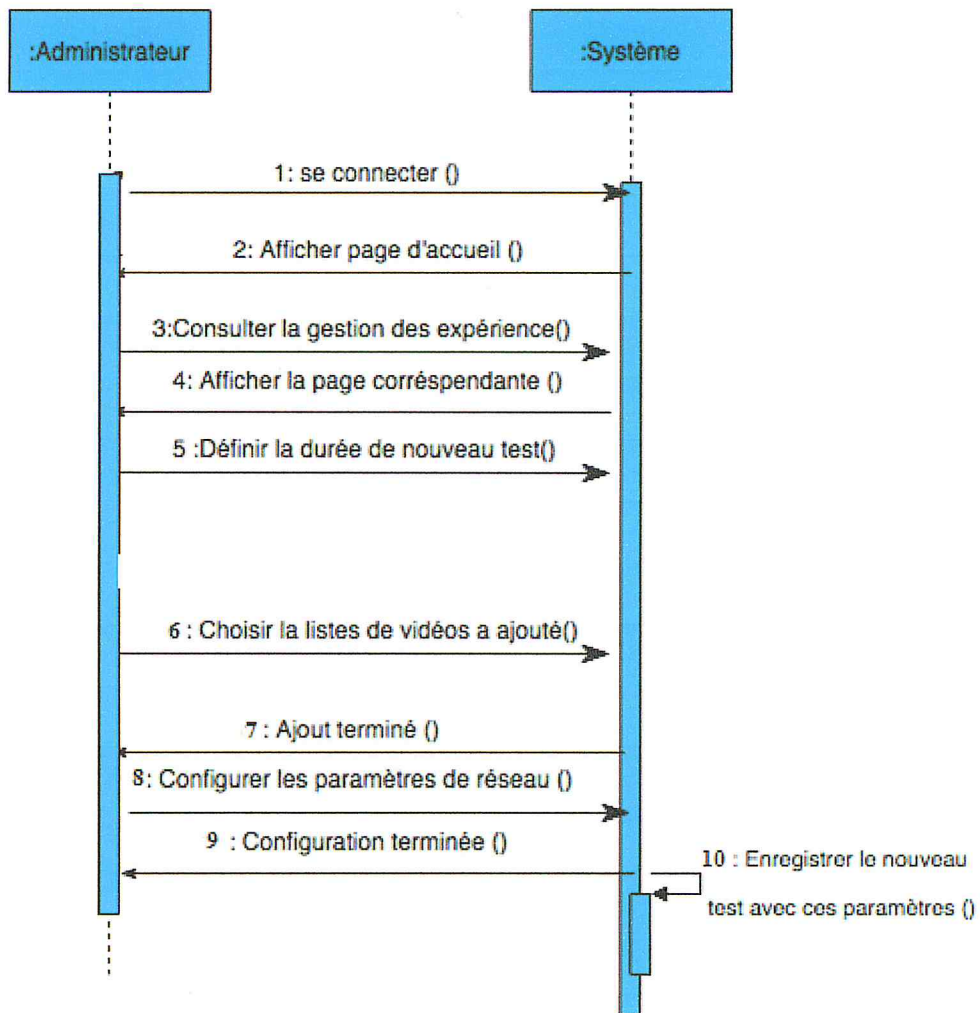


Figure 28: Diagramme de séquence de cas d'utilisation " ajout d'une expérience".

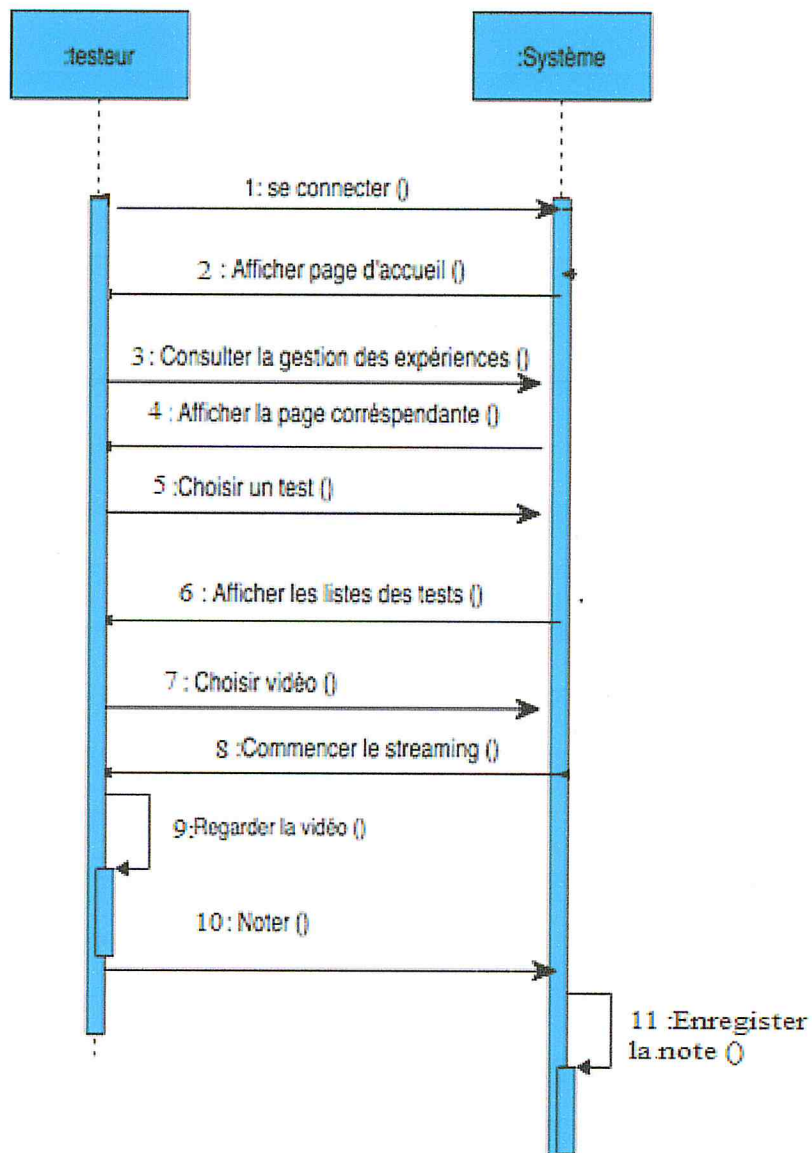
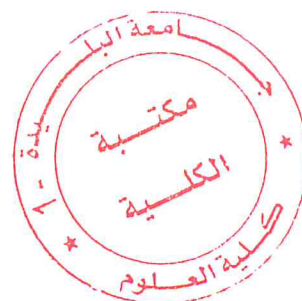


Figure 29: Diagramme de séquence de cas d'utilisation " test d'une expérience".



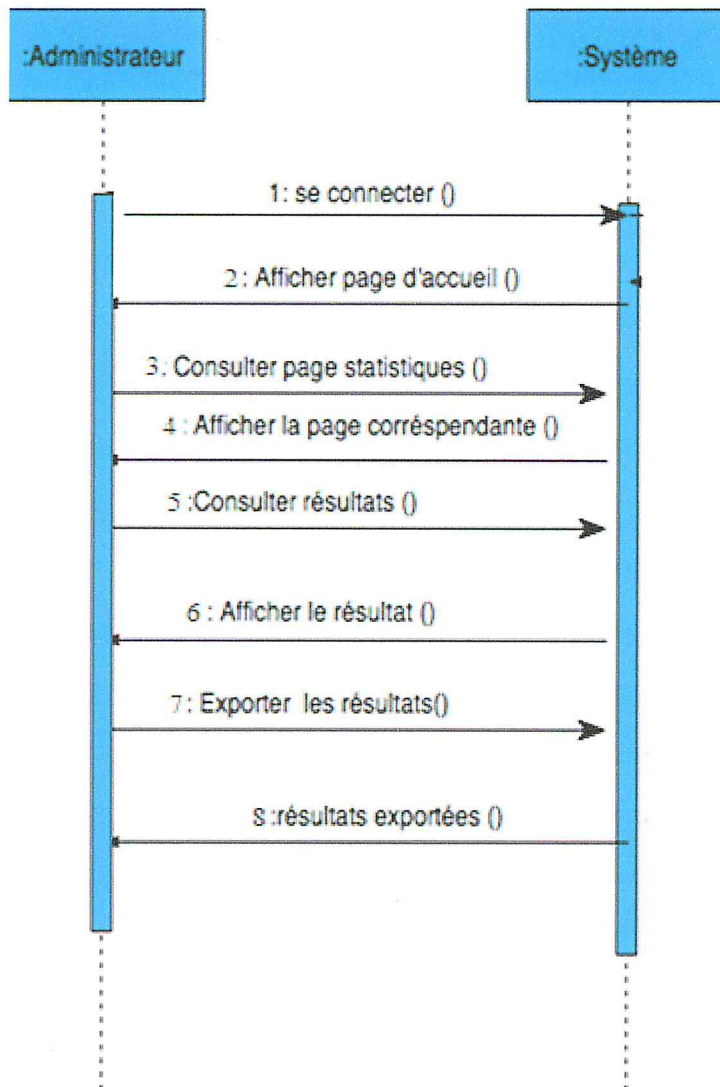


Figure 30: Diagramme de séquence de cas d'utilisation "Statistiques".

III.7 L'architecture de la plate-forme proposée :

Comme il est mentionné précédemment, notre plateforme peut fonctionner en deux environnements différents : contrôlé et réel. Dans le premier environnement, les testeurs sont connectés à un serveur local via un réseau local où ses paramètres de QoS sont contrôlés à l'aide de l'émulateur réseau. Par contre, dans un environnement réel, les testeurs sont connectés à un serveur distant via Internet, les paramètres du QoS ne sont plus contrôlables mais mesurables à l'aide de l'outil de capture de paquets. La figure 31 suivante montre l'architecture de notre plateforme.

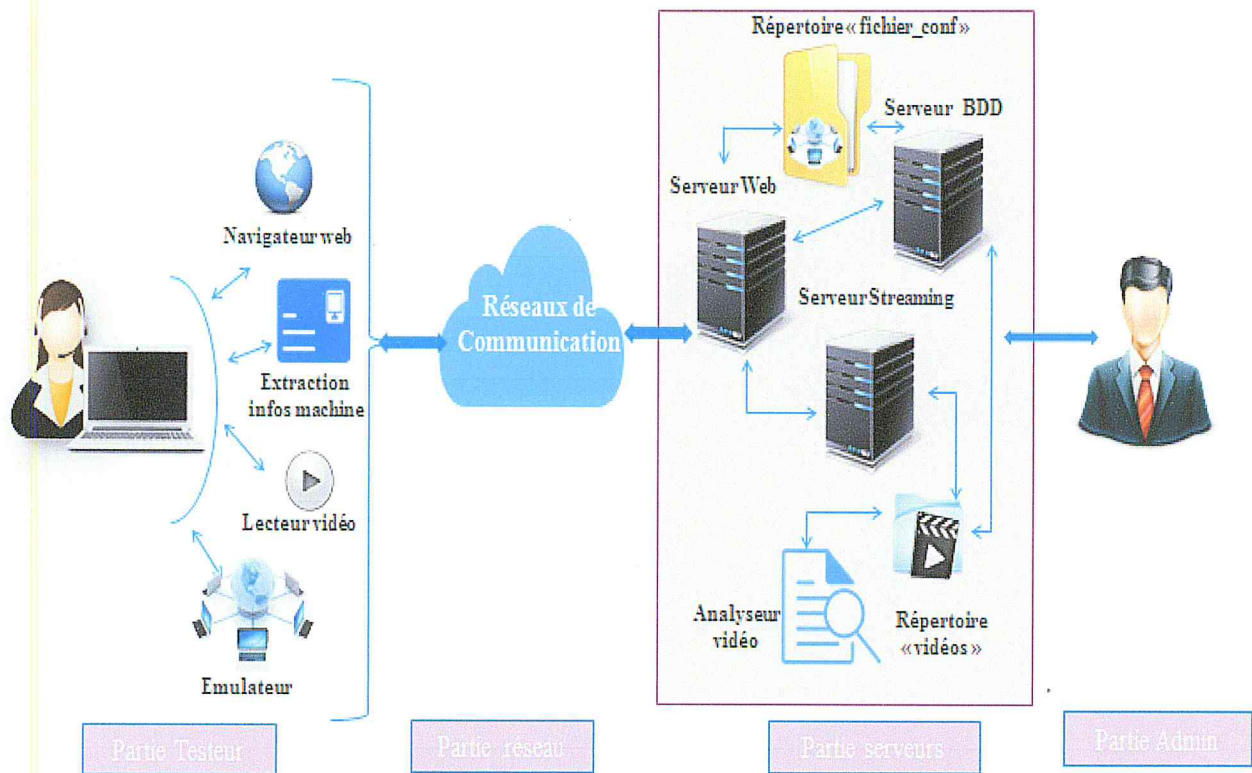


Figure 31: Architecture de la plateforme.

III.8 Notre Modèle de métriques

Comme il est cité dans le premier chapitre (section I.3.3), notre modèle de métriques inclut les facteurs suivants :

- les caractéristiques de la vidéo source comme le bitrate, la largeur et la hauteur de trame, framerate,
- Les paramètres de réseaux comme la bande passante, la latence, la gigue et la perte,
- les paramètres du terminal comme le type de l'écran, la résolution de l'écran, la capacité graphique, la puissance du processeur et la taille de RAM,
- le profil de l'utilisateur à savoir: l'âge, sexe, niveau (La fréquence d'utilisation de plateforme de streaming, Facilité d'accès à la plateforme, Débit de connexion internet que possède un utilisateur dans le cas d'un environnement réel, contexte de vidéo préféré ...)

Ces FIs sont présentés par le diagramme s'Ishikawa² (cause-effet) suivant (figure 32) :

² Le Diagramme de causes et effets, ou diagramme d'Ishikawa, ou diagramme en arêtes de poisson ou encore 5M, est un outil développé par Kaoru Ishikawa en 1962 et servant dans la gestion de la qualité. IL représente de façon graphique les causes aboutissant à un effet.

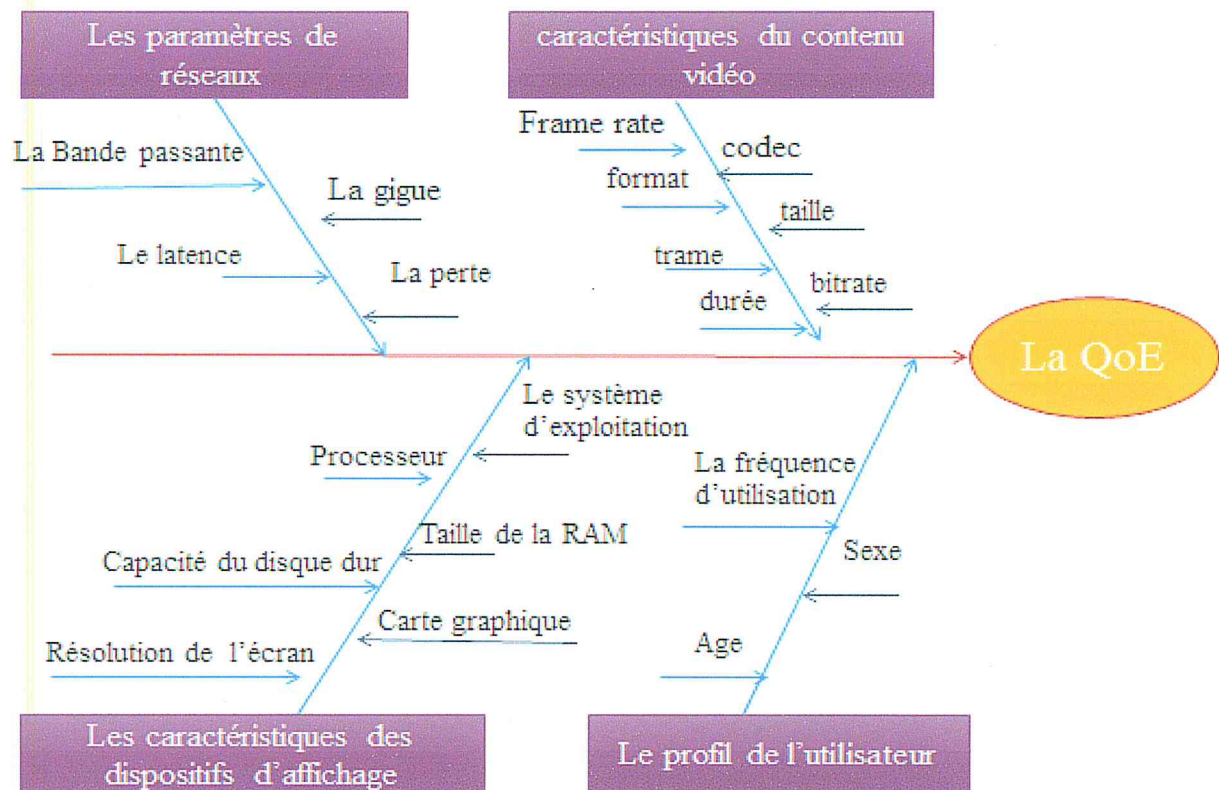


Figure 32: Diagramme d'Ishikiawa de notre modèle de métriques.

III.8.1 Le profil du testeur :

Plusieurs facteurs liés au testeur, ses préférences et son environnement peuvent influencer la QoE. Dans notre plateforme, nous nous intéressons aux paramètres suivants :

- **Age** : se traduit par la date de naissance de l'utilisateur, l'inscription peut être limité à un certaine tranche d'âge pour les besoin de l'étude menée.
- **Sexe** : pour un motif de goût en terme d'audiovisuel.
- **La fréquence d'utilisation** de plateforme de streaming.

III.8.2 Les paramètres du terminal :

On veut dire par le terminal l'ordinateur personnel (pc) sur lequel le testeur effectuera les tests. Les paramètres collectés nous permettrons d'authentifier les machines connectées à notre plateforme ainsi que les différentes caractéristiques propres à chaque machine comme :

- **Résolution de l'écran** : appelée aussi la définition d'écran, est le nombre de points ou pixels que peut afficher une carte graphique sur un écran. La définition est le produit du nombre de points selon l'horizontale par le nombre de points selon la verticale de l'affichage. Pour utiliser une définition donnée, il faut que la carte graphique de la machine soit capable de générer le signal vidéo, et que l'écran soit capable de l'afficher.
- **Capacité du disque dur**: c'est une mémoire de masse magnétique utilisée principalement dans les ordinateurs, leur choix se fait principalement en fonction de sa capacité. Les disques durs ayant les capacités les plus importantes sur le marché dépassent les 2 To (téraoctets) (2010) et 3 To en 2011. Cette capacité a augmenté beaucoup plus vite que leur rapidité, limitée par la mécanique. Le temps d'accès en

lecture est lié à la vitesse de rotation du disque et au temps de positionnement des têtes de lectures

- **Taille de RAM :** La RAM se présente sous la forme de micro puces « discrètes » (c'est-à-dire indépendantes) ou de barrettes qui s'enfichent dans des emplacements sur la carte mère de l'ordinateur. Ces emplacements se connectent au processeur par le biais d'un bus, c'est-à-dire une série de chemins électriques. Dès que la mémoire vive est pleine, le processeur doit sans cesse remplacer d'anciennes données par des nouvelles qu'il va chercher sur le disque dur, ce qui ralentit le fonctionnement de l'ordinateur. Contrairement au disque dur, qui peut devenir saturé de données au point de ne plus pouvoir en accepter de nouvelles, la RAM ne vient jamais à manquer de mémoire. Elle continue de fonctionner, mais beaucoup plus lentement, ce qui peut devenir gênant.
- **Système :** Le système d'exploitation c'est logiciel et le premier programme exécuté lors de la mise en marche de l'ordinateur, il est chargé d'assurer la liaison entre les ressources matérielles, l'utilisateur et les applications. On distingue plusieurs types de systèmes d'exploitation, selon qu'ils sont capables de gérer simultanément des informations d'une longueur de 16 bits, 32 bits, 64 bits ou plus.
- **Carte graphique :** Une carte graphique ou carte vidéo, ou encore un adaptateur graphique, est une carte d'extension d'ordinateur dont le rôle est de produire une image affichable sur un écran. La carte graphique envoie à l'écran des images stockées dans sa propre mémoire, à une fréquence et dans un format qui dépendent d'une part de l'écran branché et du port sur lequel il est branché et de sa configuration interne d'autre part.
- **Processeur :** Le processeur (ou CPU de l'anglais Central Processing Unit, « Unité centrale de traitement ») et le composant de l'ordinateur qui exécute les instructions machine des programmes informatiques. La puissance d'un processeur est le nombre d'instructions qu'il peut exécuter dans une durée donnée et ça varie d'une machine à l'autre. Le nombre de programme exécuté en même temps peut très bien ralentir le bon fonctionnement de certain programme tel que la lecture d'un flux vidéo en direct sur Internet et ce dans le cas d'un processeur qui n'est pas très puissant qui pourra générer des blocages de séquence et ainsi grandement influencer la perception du testeur.

III.8.3 Les caractéristiques de la Vidéo

La vidéo est au centre de notre travail, ainsi que l'environnement dans lequel elle sera diffusée et le périphérique sur lequel elle sera visionnée détermine le niveau de perception par les sujets humains qui seront invités à participer à notre expérience. Ses caractéristiques sont :

- **Le bitrate :** décrit la façon dont la piste vidéo ou audio est encodée. Un constant « bitrate » signifie en fait que cette piste vidéo et/ou audio utilise le même montant d'espace disque pour chaque seconde.
- **La hauteur et largeur :** La trame est le résultat de décomposition d'une image vidéo en lignes paires et lignes impaires. Chaque image est constituée ainsi de deux trames : une trame qui reproduit les lignes impaires et une deuxième les lignes paires. La décomposition de l'image vidéo en deux trames ou entrelacement est une technique

utilisée afin d'éviter l'effet de papillonnement de l'image sur un écran à faible fréquence de balayage

- **Frame rate:** c'est la Fréquence de mouvement des images.
- **Taille :** c'est la taille de la vidéo
- **Format :** Un format vidéo décrit comment un dispositif envoie des images d'une vidéo à un autre dispositif, de la même manière qu'un lecteur de DVD envoie des images à un téléviseur, ou un ordinateur à son moniteur. Plus formellement, le format visuel décrit l'ordre et la structure des images qui créent l'image vidéo.
- **Codec :** Un codec est un dispositif matériel ou logiciel permettant de mettre en œuvre l'encodage ou le décodage d'un flux de données numérique, en vue d'une transmission ou d'un stockage. Certains codecs intègrent également une fonction de compression ou encore de chiffrement des données.
- **Durée :** c'est la durée de la vidéo.

III.8.4 Les paramètres de la qualité de service (QoS)

Les paramètres de la QoS reflètent l'état du réseau et son impact direct sur la QoE. La dégradation de chaque paramètre réseau conduit à la dégradation de la qualité du streaming vidéo. Dans notre modèle, nous retenons les plus importantes :

- **La bande passante :** désigne le débit binaire maximal d'un canal de communication.
- **La Latence :** désigne le temps nécessaire à un paquet de données pour passer de la source à la destination.
- **Le taux de perte de paquet :** est le pourcentage de paquets perdus lors de la transmission de données. Le taux est une moyenne de toutes les mesures effectuées au cours d'une période donnée. Il s'agit d'un critère de qualité de services d'un réseau.
- **La gigue :** est la différence de délai de transmission de bout en bout entre des paquets choisis dans un même flux de paquets, sans prendre en compte les paquets éventuellement perdus

Ces FIs seront sauvegardé dans une BDD. Dans la section suivante, nous allons présenter le schéma rationnel de notre BDD.

III.9 Schéma relationnel de la BDD

Pour concevoir le schéma relationnel d'une base de données, nous avons effectué tout d'abord le diagramme de classe qui sera transformé vers un modèle relationnel en appliquant des règles de transformation. Cela fait l'objet de cette section

III.9.1 Diagramme de classe

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes. Outre les classes, ils représentent un ensemble d'interactions et de paquetages, ainsi que leurs relations. Les classes sont les descripteurs d'un ensemble d'objets qui ont une structure, un comportement et des relations similaires. Représentées par des rectangles compartimentés, elles sont reliées l'une à l'autre avec plusieurs formes : les associations (une classe associée une autre classe), la dépendance (une classe dépend ou utilise une autre classe), la spécialisation (une classe est une spécialisation d'une autre classe) ...etc (34).

Dans la figure 33 nous montrons notre diagramme de classe.

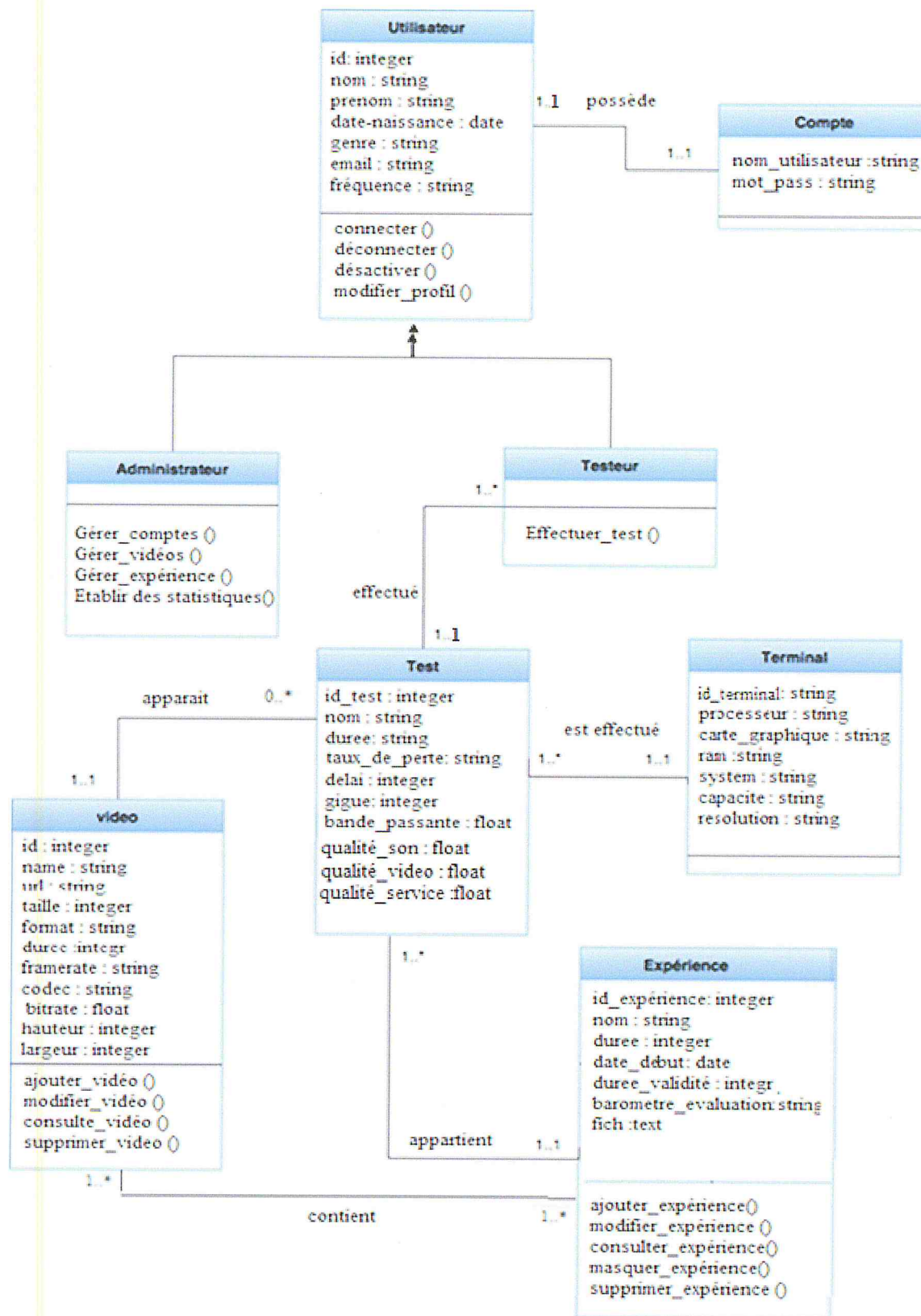


Figure 33 : Diagramme de classe de la plateforme

III.9.2 Description des classes :

Classe	Attribut	Type	Désignation	Méthode
Compte	Nom_utilisateur Mot_pass	String String	Le pseudo de l'utilisateur Le mot de passe de l'utilisateur	
Utilisateur	Id Nom Prenom Date_naissance Genre Email fréquence	Integer String String String String String String	L'identifiant du l'utilisateur Le nom de l'utilisateur Le prénom de l'utilisateur La date de naissance de l'utilisateur Le genre de l'utilisateur (femme, homme) L'adresse E-mail de l'utilisateur La fréquence d'utilisation des plateformes de streaming vidéo	Connecter() Decconnecter() Desactiver() Modifier_profil() Gérer_vidéo() Gérer_expériences() Gérer_compte() Effectuer_test() Statistiques ()
Test	Id_test nom Durée Delai Gigue Bande_passante Taux_de_perte qualité_vidéo qualité_son qualité_service	Integer String Float Float Float Float Float Float Float Float	Numéro séquentiel du test Le nom du test La durée du test Le temps nécessaire de la transmission de paquets de ce test la différence de délai de transmission de ce test le débit binaire maximal de ce test pourcentage de paquets perdus de ce test la note d'évaluation de la qualité de vidéo la note d'évaluation de la qualité de son la note d'évaluation de la qualité de service	

Vidéo	<p>id Nom url taille format durée framerate codec Bitrate Largeur Hauteur</p>	<p>Integer String String String String String String String String String String</p>	<p>L'identifiant de la vidéo Le nom de la vidéo qui donne une information sur le contenu L'adresse url de la vidéo où bien l'emplacement La taille de la vidéo décrit l'ordre et la structure des images qui créent l'image vidéo. La durée de la vidéo La fréquence de mouvement de l'image de vidéo Le dispositif matériel ou logiciel permettant de mettre en oeuvre l'encodage ou le décodage de la vidéo La façon dont la piste vidéo est encodée Largeur de la trame de cette vidéo La hauteur de la trame de cette vidéo</p>	<p>Ajouter_vidéo() Modifier_vidéo() Supprimer_vidéo() Consulter_vidéo()</p>
Expérience	<p>Id_expérience Nom Durée Date_debut Duree_validité Barometre_evaluation Fiche</p>	<p>Integer String Integer date string string string TEXT</p>	<p>L'identifiant de l'expérience Le nom de l'expérience La durée fixée de cette expérience La date de début de création de l'expérience La durée de validation de l'expérience Le type d'évaluation (continue ou discret) Le fichier de configuration de réseau</p>	<p>Ajouter_expérience Modifier_expérience() Supprimer_expérience() Rechercher_expérience() Masquer_expérience()</p>

Terminal				
Id_terminal	String	L'identifiant de la machine		
Processeur	String	Le type de processeur		
Carte_graphique	String	La capacité graphique de la machine		
Ram	String	La taille de la mémoire RAM de la machine		
System	string	Le système d'exploitation de la machine		
Capacité	string	La capacité du disque dur		
Résolution	string	La résolution d'image de la machine		

Tableau 12 : Tableau descriptif des classes.

III.9.3 Dictionnaire des données des relations

Realtion	Collection	Identifiant	Cardinalité
Possede	Compte	Id_compte	1..1
	Utilisateur	Id	1..1
effectuer	Testeur	Id	1..*
	Test	Id_test	1..1
Est effectué	Test	Id_test	1..*
	Terminal	Id_terminal	1..1
Executer	Test	Id_test	1..*
	Terminal	Id_terminal	1..*
appartient	Test	Id_test	1..*
	expérience	Id_expérience	1..1
Contient	Expérience	Id_expérience	1..*
	vidéo	Id	1..*
apparaît	Test	Id_test	0..*
	Vidéo	Id	1..1

Tableau 13: Dictionnaire des données des relations.

III.9.4 Passage au modèle relationnel

Les règles de passage au modèle relationnel sont :

- **Transformation des classes** : chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé.
- **Transformation des associations** : Nous distinguons trois familles d'associations
 - **Association 1.. ***: il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

- **Association *.* et n-aire et classes-association** : la classe-association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des classes connectées à l'association.
- **Association 1.. 1** : il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association. Si les deux multiplicités minimales sont à un, il est préférable de fusionner les deux classes en une seule.

III.9.5 Le schéma relationnel

En appliquant ces règles de transformation d'un diagramme de classe vers un modèle relationnel, nous avons abouti au schéma relationnel suivant (figure34):

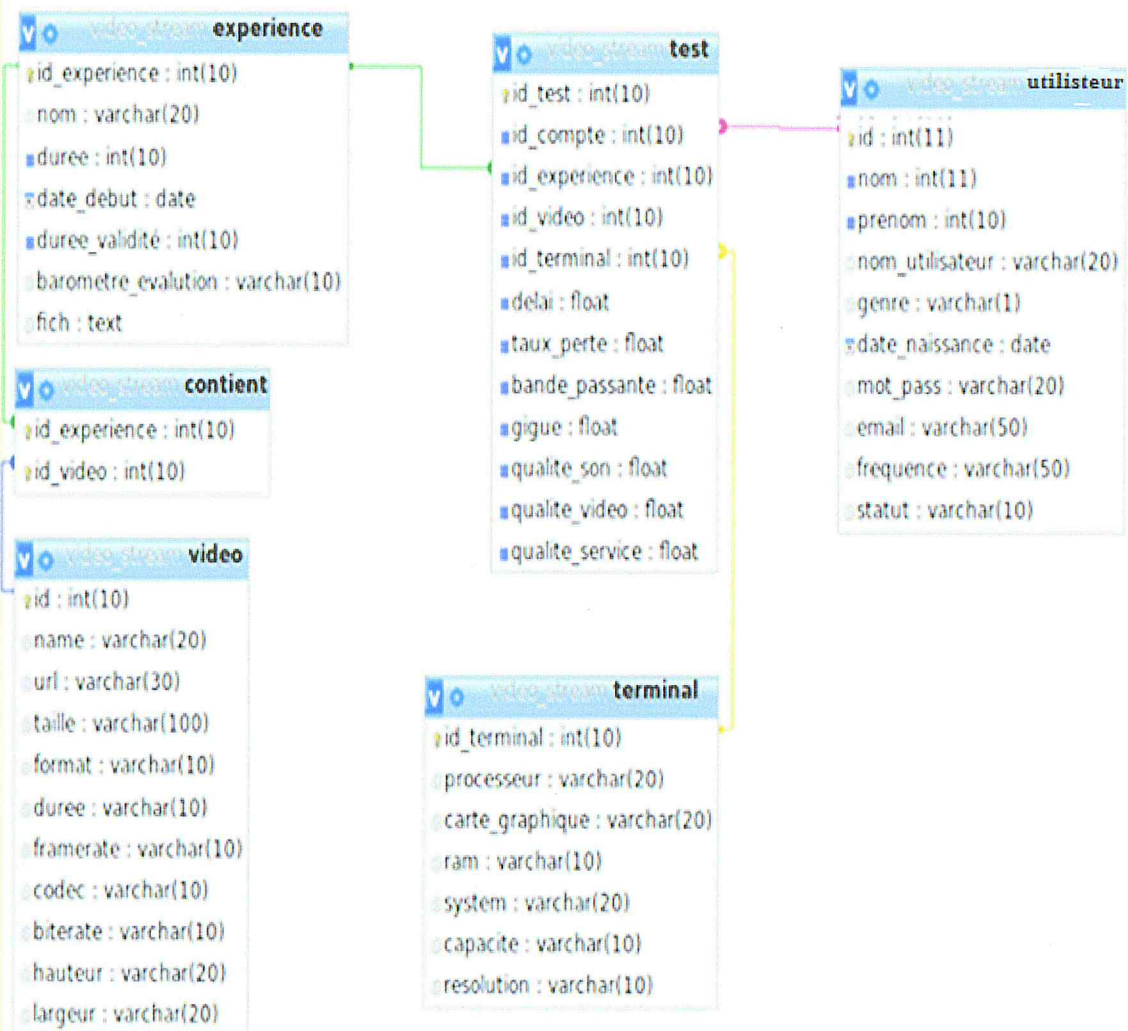


Figure 34 : Le schéma relationnel de la base des données.

III.10 Le processus de collection d’informations :

Le processus de collecte d’information est constitué d’une multitude de fonctions réparties sur chaque étape de l’utilisation de l’application (Tableau 14):

- a. à l’inscription où les informations liées au profil du testeur sont enregistrées,
- b. à la connexion où sont collectées les caractéristiques du terminal du testeur,
- c. à lancement du test où sont enregistrés les paramètres QoS (dans le cas d’environnement contrôlé).
- d. à la fin du test où les trois notes de la qualité sont enregistrées. Dans le cas de l’environnement réel, les paramètres QoS sont aussi enregistrés.

Nous tenons à préciser que les caractéristiques de la vidéo seront enregistrées à l’ajout dans la plateforme

FIs	Etape de collecte
Profil du testeur	Inscription
Caractéristiques de la vidéo	Ajout d’une vidéo par l’administrateur.
Caractéristiques du terminal	Connexion (authentification) du testeur
Paramètres QoS	Contrôlé : Lancement d’un test Réel : Achèvement d’un test
Evaluation	Achèvement du test

Tableau 14: FIs collectées à chaque étape.

Dans la section qui suit, nous montrons la modélisation de ce processus de collecte d’informations par le diagramme de séquence (figure 35 et figure 36)

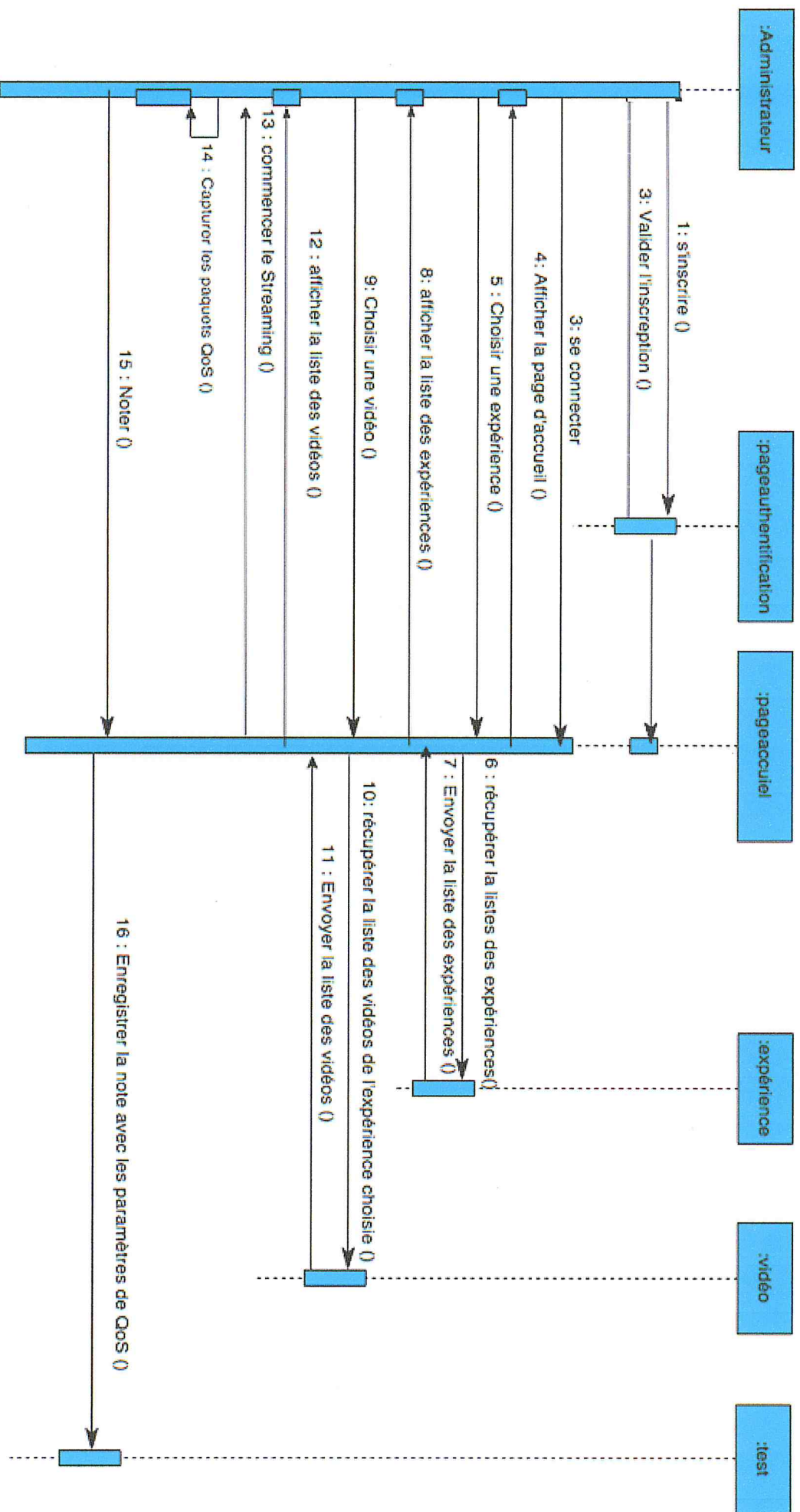


Figure 35: Processus de collecte d'information cas " environnement réel".

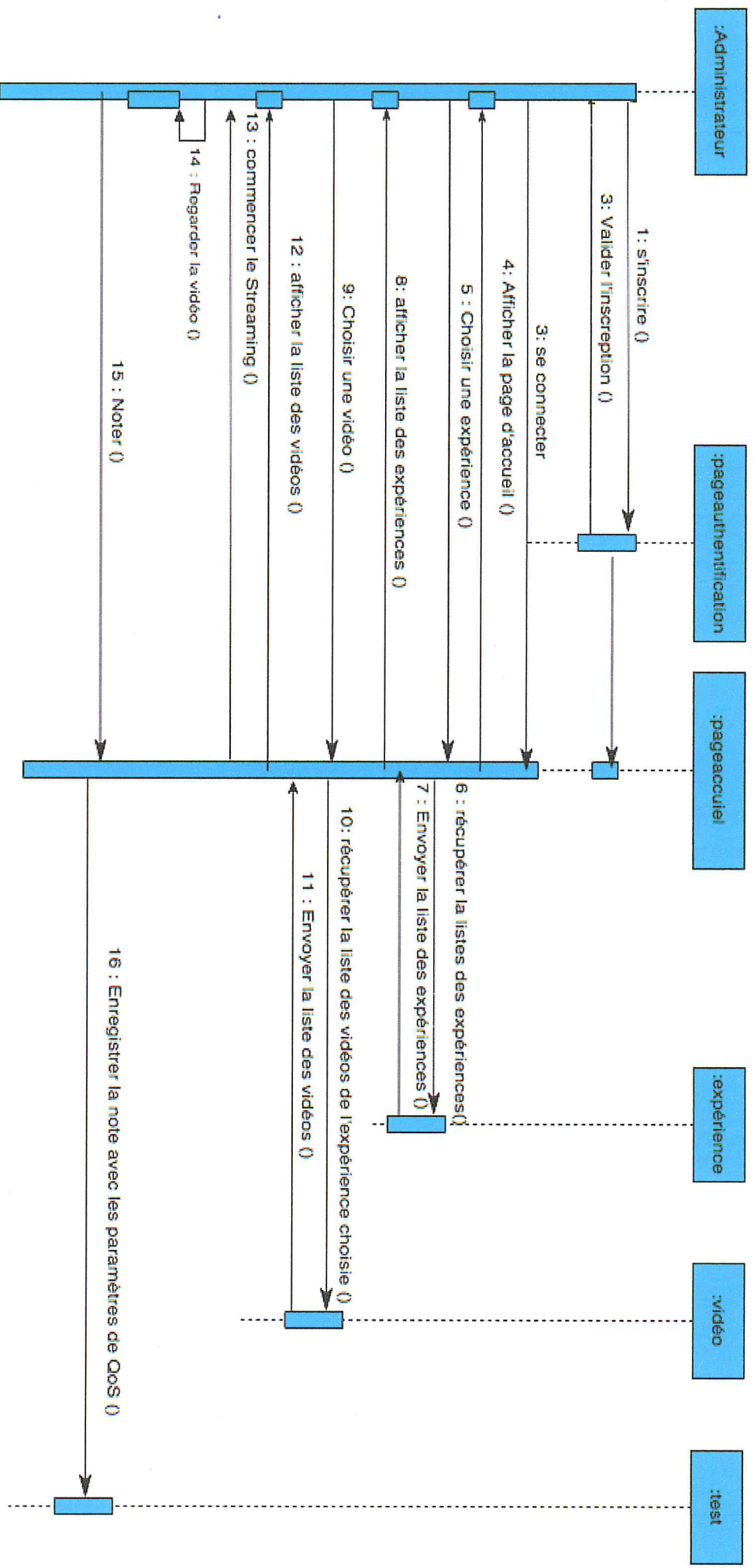


Figure 36 : Processus de collecte d'information cas "environnement contrôlé".

III.11 Conclusion

Dans ce chapitre, nous avons spécifié les fonctionnalités de notre plateforme que nous avons modélisées avec les diagrammes UML. L'objectif de ce chapitre été de proposer une solution conceptuelle qui répond aux besoins définis lors de la phase d'analyse. Elle permet de spécifier une solution technique qui sera concrétisée par la suite. Dans le chapitre suivant nous montrerons les étapes, plus en détails, que nous avons suivies pour implémenter et réaliser notre solution.

Chapitre 4 : Implémentation

IV.1 Introduction

Après avoir achevé l'étape de conception de la plateforme, nous allons entamer dans ce chapitre la partie implémentation qui constitue le dernier volet de ce mémoire et qui a pour objectif d'exposer notre plateforme « **QoEStreamingVideo** », une plateforme d'expérimentation pour évaluer la Qualité d'Expérience des Utilisateurs des Services de Streaming vidéo. Pour ce faire, nous allons commencer tout d'abord par préciser l'environnement matériel et logiciel de travail. Ensuite, nous couvrirons les différents langages de programmation, et technologies utilisées. Enfin nous présenterons quelques interfaces de notre plateforme.

IV.2 L'environnement de développement

Un **environnement de développement** se réfère à une suite d'applications et d'outils que les développeurs ont installé sur leurs machines les aidant à développer leurs applications, gérer les fichiers sources, déboguer du code, et enfin tester leur travail avant de le lancer sur un environnement de test et/ou de production. La figure 37 suivante montre l'architecture de la plateforme QoEStreamingVideo ainsi nous détaillerons par la suite les outils utilisés.

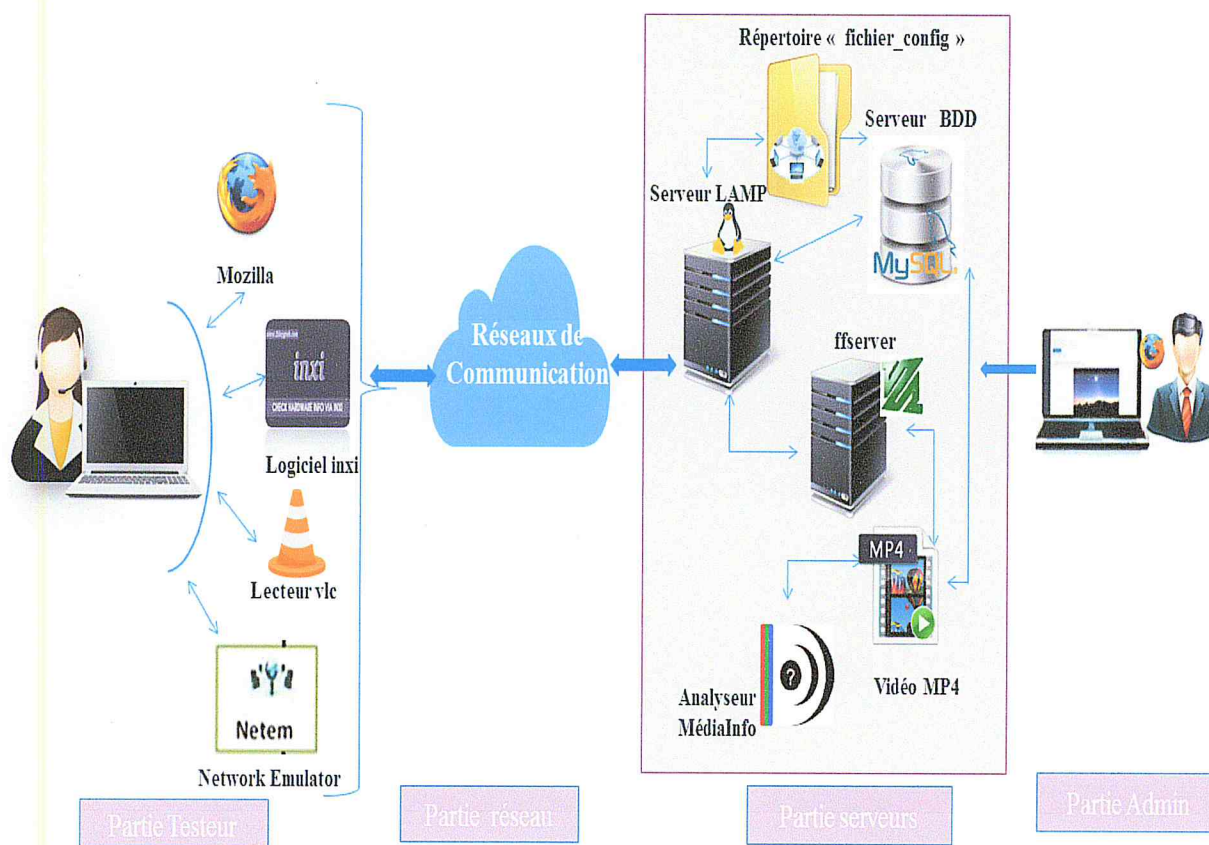


Figure 37: L'architecture globale de la plateforme QoEStreamingVideo.

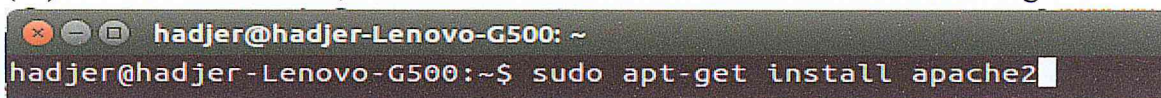
IV.2.1 Outils de développement

Lors du développement de notre plateforme, on a utilisé les outils suivants :

IV.2.1.1 LAMP

LAMP (acronyme de Linux Apache MySQL PHP) qui est une suite de logiciels open source constituée d'un :

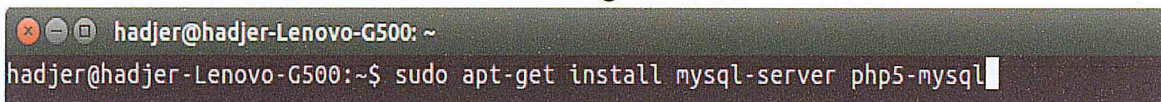
- **Système d'exploitation** (Linux.) dans notre cas nous avons choisir de travailler avec **Ubuntu** C'est un système d'exploitation libre Fondé sur la distribution Linux Debian, ce système d'exploitation est constitué de logiciels libres, et est disponible gratuitement, y compris pour les entreprises (35). Dans notre cas nous avons travaillé avec la version 14.04.
- **Serveur Apache** (serveur HTTP) : c'est un serveur HTTP en Open Source utilisé principalement sur les hébergements Internet en Linux, bien qu'il soit également utilisable en Windows (concurrent d'Internet Information Service - IIS), Unix ou OS X. C'est actuellement le plus utilisé sur le WEB. Différentes fonctionnalités sont implantées comme la possibilité d'utiliser un seul serveur Internet pour héberger plusieurs sites, l'utilisation des langages interprétés Perl, PHP et Python, sauvegarde des accès dans un fichier log (statistiques), Htaccess pour la protection des répertoires et l'URL-Rewriting, ... (36). Pour son installation, nous avons exécuté la commande illustrée dans la figure 38.



```
hadjer@hadjer-Lenovo-G500: ~  
hadjer@hadjer-Lenovo-G500:~$ sudo apt-get install apache2
```

Figure 38:l'installation de serveur apache2 sous ubuntu.

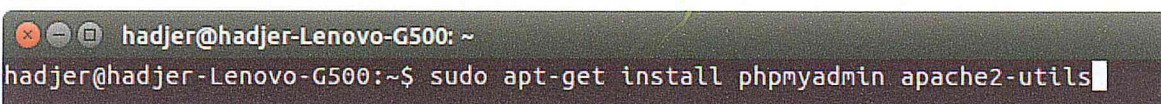
- **Serveur MySQL** : c'est un Système de Gestion des Bases de données (SGBD) Open Source, très rapide, robuste et multiutilisateur, pour faire simple, son rôle est d'enregistrer des données de manière organisée afin de nous aider à les retrouver facilement plus tard. Le serveur MySQL supporte le langage de requêtes SQL : le langage standard de choix des SGBD modernes. La portabilité du serveur MySQL lui permet de s'exécuter sur toutes les plateformes et d'être intégré à plusieurs serveurs web (37). Pour son installation, nous avons exécuté la commande illustrée dans la figure 39.



```
hadjer@hadjer-Lenovo-G500: ~  
hadjer@hadjer-Lenovo-G500:~$ sudo apt-get install mysql-server php5-mysql
```

Figure 39: l'installation de serveur MySQL sous ubuntu

- **Interface PhpMyAdmin** : c'est une application web qui permet de gérer un serveur de base de données MySQL. Dans un environnement multiutilisateur, cette interface écrite en PHP permet également de donner à un utilisateur un accès à ses propres bases de données. Pour son installation, nous avons exécuté la commande illustrée dans la figure 40.



```
hadjer@hadjer-Lenovo-G500: ~  
hadjer@hadjer-Lenovo-G500:~$ sudo apt-get install phpmyadmin apache2-utils
```

Figure 40: l'installation de PhpMyAdmin sous ubuntu.

IV.2.1.2 Les langages d'implémentation utilisés

Après avoir bien choisi l'environnement où on va développer notre plateforme, maintenant on passe aux langages d'implémentation utilisés. On est amenés à manipuler non pas un mais plusieurs langages. Quel que soit le site web que l'on souhaite créer, **HTML** et **CSS** sont indispensables, cependant, ces langages ne suffisent pas pour créer un site dynamique, il faut compléter ces deux langages avec d'autres, dans notre cas on a utilisé le langage de script **PHP** ainsi que le langage de requêtes **SQL**.

- **Le langage HTML (HyperText Markup Language) :** c'est le langage de création des sites web, il a fait son apparition dès 1991 lors du lancement du Web, son rôle est de gérer et organiser le contenu, c'est donc en HTML que on écrit ce qui doit être affiché sur les pages de notre plateforme : du texte, des liens, des images ...etc (38).
- **CSS (Cascading Style Sheets ou bien Feuilles de style) :** Ce langage est venu compléter le HTML en 1996, c'est un langage de mise en forme des sites web. Tandis que le HTML permet d'écrire le contenu de pages web et de les structurer, le langage CSS s'occupe de la mise en forme et de la mise en page, il gère l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...etc.) (39).
- **Bootstrap :** Une boîte à outils HTML, CSS et JavaScript destinés à simplifier la création d'applications web. En août 2011, Twitter lançait Bootstrap. Ce dernier inclut un système de grille, des styles par défaut pour les principaux types de contenus, mais aussi le code nécessaire pour faire fonctionner les composants les plus courants qu'on retrouve sur des sites web (popups modales, messages d'alertes, carrousel, etc...).
- **Le langage de script PHP :** C'est un langage de programmation que seuls les serveurs comprennent et qui permet de rendre le site dynamique, c'est PHP qui génère la page web, ce langage peut fonctionner seul mais il ne prend vraiment de l'intérêt que s'il est combiné à un outil tel que MySQL. PHP est utilisé sur de nombreux serveurs pour prendre des décisions, c'est lui qui décide du code HTML qui sera généré et envoyé au client à chaque fois. Pour son installation, nous avons exécuté la commande illustrée dans la figure 41.



```
hadjer@hadjer-Lenovo-G500: ~
hadjer@hadjer-Lenovo-G500:~$ sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
```

Figure 41: l'installation de PHP sous ubuntu

Vu qu'on ne peut pas éditer un code sans avoir utilisé un éditeur text, pour cela nous avons utilisé :

- **Sublime Text** qui est un éditeur de texte générique codé en C++ et Python, il est disponible pour plusieurs systèmes d'exploitation tel que Windows, Mac et Linux.il supporte plusieurs langages dont PHP.
- **Le langage des requêtes SQL :** C'est un langage qui permet en effet d'effectuer des calculs directement sur les données à l'aide de fonctions toutes prêtes, ces fonctions sont spécialement dédiées aux bases de données, c'est **SQL** qui permet de communiquer avec la base de données. Ce langage est un standard, c'est-à-dire que quel que soit le **SGBD** qu'on utilise (MySQL, PostgreSQL, SQLite, Oracle ou Microsoft SQL Server), on utilisera **SQL** pour donner des ordres au SGBD.

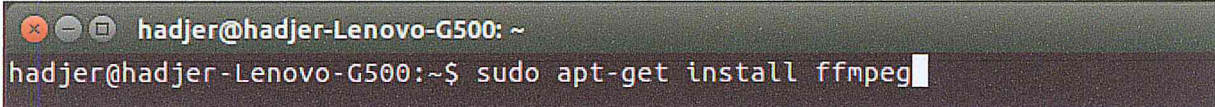
Pour faire le streaming Nous avons utilisé aussi les outils **FFmpeg & FFserver**, **VLC** comme un lecteur ainsi **MediaInfo** pour extraire les caractéristiques des vidéos et **INXI** pour extraire les informations de la machine. Dans qui suit nous détaillons chacun de ces outils.

IV.2.1.3 Les outils d'implémentation de Streaming

La réalisation de Streaming nous a amené dans un premier temps à installer **FFmpeg** (40) qui est une collection de logiciels libres destinés au traitement de flux audio ou vidéo. Il est constitué de plusieurs composants outils :

- en ligne de commande :
 - **ffmpeg** : l'utilitaire éponyme pour convertir des formats vidéo. Il supporte aussi l'encodage en temps réel depuis une carte TV ;
 - **ffserver** est un serveur HTTP destiné à la diffusion de contenu en continu (streaming) ;
 - **ffplay** pour lire un fichier multimédia (il est basé sur les bibliothèques SDL et FFmpeg).
- Ou Bibliothèques logicielles:
 - **libavcodec** contient tous les encodeurs et décodeurs audio/vidéo de FFmpeg. La plupart des codecs ont été redéveloppés à partir de zéro pour assurer les meilleures performances et la réutilisabilité du code source, une partie d'entre eux ont été développés par rétro-ingénierie ;
 - **libavformat** contient un analyseur syntaxique (un parser en anglais, parcourreur en français) et un générateur pour les formats audio/vidéo les plus communs.

Pour son installation, on a exécuté la commande suivante en mode sudo (figure 42) :



```
hadjer@hadjer-Lenovo-G500: ~  
hadjer@hadjer-Lenovo-G500:~$ sudo apt-get install ffmpeg
```

Figure 42: l'installation de FFmpeg sous ubuntu.

Dans notre cas, on avait besoin d'un serveur de diffusion, dont l'un est **ffserver** (41) qui est parmi les composants de **ffmpeg** cités en dessus. **FFserver** est un serveur de streaming pour l'audio et la vidéo. il fonctionne en transférant des flux codés par **ffmpeg**, ou des flux préenregistrés qui sont lus à partir du disque. Vous pouvez rechercher des positions dans le passé sur chaque flux en direct, à condition de spécifier un stockage d'alimentation suffisamment grand. Il reçoit des fichiers préenregistrés ou des flux **FFM** provenant d'une instance **ffmpeg** comme entrée, puis les diffuse sur **RTP / RTSP / HTTP**.

Il fonctionne en transférant des flux codés par **ffmpeg** ou des flux préenregistrés qui sont lus à partir du disque.

Pour pouvoir démarrer **ffserver** avec succès, vous devez d'abord disposer d'un fichier de configuration (figure 43) localisé dans le répertoire **/etc/ffserver.conf**.

```
HTTPPort 8090
RTSPPort 5454
#HTTPBindAddress 192.168.8.100
HTTPBindAddress 0.0.0.0
MaxHTTPConnections 2000
MaxClients 1000
MaxBandwidth 1000
CustomLog -
#NoDaemon

<Stream file1.mp4>
Format rtp
File "/home/hadger/Bureau/VQE/Videos/mp4/video1.mp4"
</Stream>

<Stream file2.mp4>
Format rtp
File "/home/hadger/Bureau/VQE/Videos/mp4/video2.mp4"
</Stream>

<Stream file3.mp4>
Format rtp
File "/home/hadger/Bureau/VQE/Videos/mp4/video3.mp4"
</Stream>
```

Figure 43: Exemple d'un fichier de configuration de ffmpeg

Une fois que nous avons créé un fichier de configuration valide, nous avons pu lancer ffmpeg simplement en exécutant la commande suivante (figure44):

```
hadger@hadger-Lenovo-G500:~$ ffmpeg -f /etc/ffmpeg.conf
```

Figure 44: le lancement de ffmpeg.

Et à la fin on a pu lire les vidéo diffusées à travers le lecteur **VLC** qui est un lecteur multimédia libre, gratuit et multiplateforme (Windows, GNU/Linux, BSD, Mac OS X, Pocket PC) Il supporte les codecs nécessaires à la lecture de la plupart des formats audio et vidéo. Il permet aussi de diffuser un contenu multimédia (vidéo, audio) à travers un réseau local ou le réseau Internet (42). Pour se faire nous avons installé VLC dans un premier temps à travers la commande suivante (figure 45).

```
hadger@hadger-Lenovo-G500:~$ apt-get install vlc
```

Figure 45: l'installation de VLC sous ubuntu.

Après nous avons ouverts un flux de réseau à partir de la barre Menu de lecteur VLC, tout en tapant la ligne suivante (figure 46).

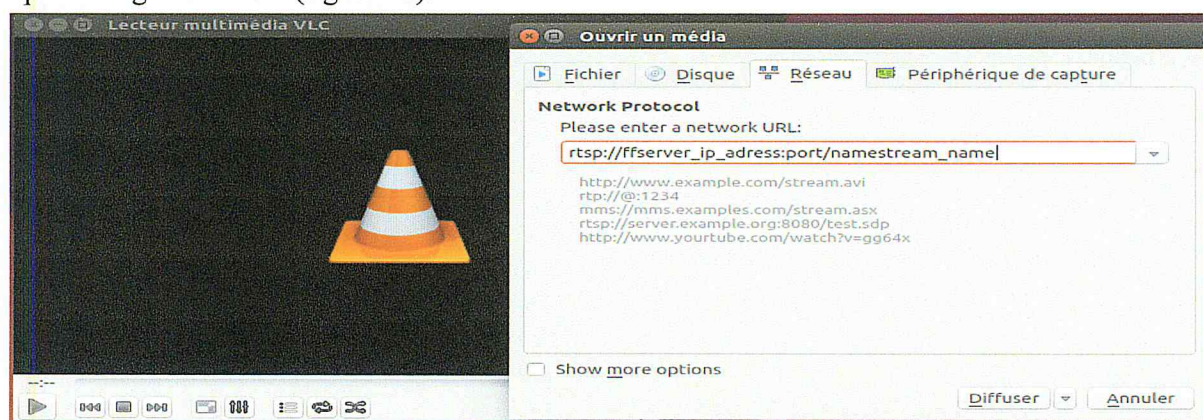


Figure 46 : la configuration d'un flux de réseau du lecteur VLC.

Où `ffserver_ip_address` est l'adresse IP de la machine où `ffserver` est installé, `port` est le numéro de port du serveur et `namestream_name` est le nom de l'alimentation correspondante définie dans le fichier de configuration.

Après avoir configuré le serveur VLC pour une lecture d'un fichier multimédia, les clients peuvent ensuite accéder à ce contenu en ouvrant un flux réseau à partir de VLC dans l'espace client.

Remarque : On a intégré VLC dans le code Php, cela permis la lecture de Streaming dans une page web. Pour ce faire, on a spécifié un lien vers le fichier pour que VLC s'exécute à travers la commande suivante (figure 47) :

```
<?php
echo "<center><embed type='application/x-vlc-plugin' name='video'
autoplay='yes' loop='yes' width='600' height='350'
target='rtsp://127.0.0.1:5454/' $name_video .'.mp4' /></center> ";
?>
```

Figure 47: L'intégration de VLC avec Php.

IV.2.1.4 Les outils d'extraire des informations :

- **De la vidéo :** pour extraire les caractéristiques liées aux vidéos nous avons choisi le logiciel **MédiaInfo (43)** : c'est un logiciel qui fournit des informations techniques sur un fichier vidéo et fournit par la même occasion des informations sur les paramètres audio du fichier. Parmi ces informations, on peut citer le titre ou le nom de la vidéo ou encore des informations techniques telles que le codec (MPEG par exemple), le Frame Rate, longueur et largeur de la trame vidéo etc. on a pu l'installer à travers la commande illustrée dans la figure 48.

Figure 48: l'installation de MédiaInfo sous ubuntu.

Voici un exemple du résultat obtenu en exécutant MediaInfo dans le terminal (figure 49).

```

hayet@hayet-Lenovo-Ideapad-100-15IBD:~$ mediainfo /var/www/html/video_stream/client/video/video4.mp4
General
Complete name      : /var/www/html/video_stream/client/video/video4.mp4
Format             : Flash Video
File size          : 483 KiB
Duration           : 41s 425ms
Overall bit rate   : 95.5 Kbps
Writing application : Lavf57.71.100
copyright          : Editions ENI

Video
Format             : Sorenson Spark
Codec ID           : 2
Duration           : 41s 425ms
Bit rate           : 195 bps
Width              : 320 pixels
Height             : 240 pixels
Display aspect ratio : 4:3
Frame rate mode    : Constant
Frame rate         : 15.000 fps
Bit depth           : 8 bits
Bits/(Pixel*Frame) : 0.000
Stream size        : 1 009 Bytes (0%)

Audio
Format             : MPEG Audio
Format version     : Version 1
Format profile     : Layer 3
Codec ID           : 2
Codec ID/Hint      : MP3
Duration           : 41s 404ms
Bit rate mode      : Constant
Bit rate           : 64.0 Kbps
Channel(s)         : 1 channel
Sampling rate      : 44.1 KHz
Compression mode   : Lossy
Delay relative to video : -25ms
Stream size        : 323 KiB (67%)

```

Figure 49 : Le résultat obtenu avec Médiainfo sous ubuntu.

MediaInfo offre la possibilité d'extraire les informations voulues à partir d'un fichier texte de sortie ainsi de pouvoir collecter les informations de nos vidéo participante aux teste à travers la commande suivante (figure 50).

```
mediainfo '/repertoire/fichiermedia.mp4' > '/destination/nomdufichier.txt'
```

Figure 50 : Collecter les informations d'une vidéo avec mediaInfo.

Pour cela, Nous avons intégré cet outil dans Php à travers la commande suivante (figure 51) :

```

287     move_uploaded_file($substrite, "video/" . $sname);
288     $file="/var/www/html/video_stream/client/video/$sname";
289     exec ("/usr/bin/mediainfo" . $file . " >info.txt");
290
291     $fichier = file('info.txt');
292     $pos=explode(":", $fichier[5]);
293     $taille=$pos[1];
294     $posi=explode(":", $fichier[2]);
295     $format=$posi[1];
296     $posit=explode(":", $fichier[6]);
297     $duree=$posit[1];
298     $positi=explode(":", $fichier[32]);
299     $framerate=$positi[1];
300     $position=explode(":", $fichier[26]);
301     $bitrate=$position[1];
302     $po=explode(":", $fichier[29]);
303     $hauteur=$po[1];
304     $p=explode(":", $fichier[28]);
305     $largeur=$p[1];

```

Figure 51: L'intégration de media Info sous Php.

- **De la machine** : pour extraire les informations liées à la machine connectée nous avons utilisé le logiciel **Inxi (44)** : est un ensemble de scripts qui permettront de détecter une foule d'informations sur le matériel, y compris les détails du fournisseur, la configuration du pilote de périphérique, etc. Et surtout, il imprimera tout dans un format facile à lire. Pour son installation, nous avons exécuté la commande illustrée dans la figure 52.

```
hadjer@hadjer-Lenovo-G500: ~
hadjer@hadjer-Lenovo-G500:~$ sudo apt-get install inxi
```

Figure 52: l'installation d'INXI sous ubuntu.

Inxi dispose d'une énorme liste d'options qui permettront d'afficher de plus en plus d'informations sur différentes pièces matérielles. Donc, dans notre cas, nous allons couvrir quelques éléments de base dont on a besoin .Pour cela, le drapeau b présentera une bonne quantité d'informations de base sur le système la figure 53 montre son résultat.

```
hayet@hayet-Lenovo-Ideapad-100-15IBD:~$ inxi -F
System:   Host: hayet-Lenovo-Ideapad-100-15IBD Kernel: 4.8.0-36-generic x86_64 (64 bit) Desktop: Unity 7.4.0
          Distro: Ubuntu 16.04 xenial
Machine:  System: LENOVO (portable) product: 80QQ v: Lenovo ideapad 100-15IBD
          Mobo: LENOVO model: Nano 5B6 v: NO DPK BIOS: LENOVO v: E0CN16WH date: 08/04/2015
CPU:      Dual core Intel Core i3-5005U (-HT-MCP-) cache: 3072 KB
          clock speeds: max: 1900 MHz 1: 1900 MHz 2: 1900 MHz 3: 1900 MHz 4: 1899 MHz
Graphics: Card: Intel Broadwell-U Integrated Graphics
          Display Server: X.Org 1.18.4 drivers: (unloaded: fbdev,vesa) Resolution: 1366x768@60.05hz
          GLX Renderer: Mesa DRI Intel HD Graphics 5500 (Broadwell GT2) GLX Version: 3.0 Mesa 17.0.7
Audio:    Card-1: Intel Wildcat Point-LP High Definition Audio Controller driver: snd_hda_intel
          Card-2: Intel Broadwell-U Audio Controller driver: snd_hda_intel
          Sound: Advanced Linux Sound Architecture v: k4.8.0-36-generic
Network:  Card-1: Realtek RTL8101/2/6E PCI Express Fast/Gigabit Ethernet controller driver: r8169
          IF: enp2s0 state: down mac: 50:7b:9d:6c:81:a8
          Card-2: Realtek RTL8723BE PCIe Wireless Network Adapter driver: rtl8723be
          IF: wlp3s0 state: down mac: b0:c0:90:2a:45:c2
Drives:   HDD Total Size: 500.1GB (2.4% used) ID-1: /dev/sda model: ST500LT012 size: 500.1GB
Partition: ID-1: / size: 88G used: 7.7G (10%) fs: ext4 dev: /dev/sda7
          ID-2: swap-1 size: 4.21GB used: 0.00GB (0%) fs: swap dev: /dev/sda6
RAID:     No RAID devices: /proc/mdstat, md_mod kernel module present
Sensors:  System Temperatures: cpu: 50.0C mobo: N/A
          Fan Speeds (in rpm): cpu: N/A
Info:     Processes: 244 Uptime: 7:11 Memory: 1953.9/3871.0MB Client: Shell (bash) inxi: 2.2.35
hayet@hayet-Lenovo-Ideapad-100-15IBD:~$
```

Figure 53: le résultat obtenu avec inxi.

IV.2.1.5 Module réseau

Dans cette partie nous allons parler de l'outil d'émulation de réseau utilisé lors de développement de notre plateforme.

NeTem (45) :

Dans le cadre d'une expérimentation laboratoire, le rôle d'un émulateur réseau est d'imiter les fonctions d'un réseau étendu. Nous avons choisi l'émulateur réseau Network Emulator ou Netem parmi de nombreux logiciels tels que dummynet, NIST Net ou encore ALTQ pour différente raison la principale c'est que Netem est gratuit et il est open source.

Netem émule un réseau étendu en donnant la possibilité de réguler soi-même les paramètres de ce réseau comme le délai, la perte de paquets, la duplication de paquets, la corruption de

paquets, le ré-ordonnancement de paquets et le contrôle de débit. Par ailleurs, il permet aussi de spécifier l'interface réseau à travers laquelle le trafic sera filtré en entrée et/ou en sortie. Et pour donner un exemple des commandes Netem la figure suivante montre comment simuler des paramètres réseau tel que le taux de perte de paquet ou le délai etc.

```
#tc qdisc add dev enp2s0 root netem delay 200ms loss1%
```

Figure 54: exemple commande Netem sur les paquets émis

La commande précédente simule un délai fixe et régulier de 200 millisecondes et un Taux de perte de paquet équivalent à 1%. A noter que la commande exposée dans la figure 53 génère les paramètres du délai et du Taux de perte sur les paquets émis par l'interface enp2s0

Un autre exemple d'instruction Netem mais qui permet cette fois de générer des perturbations sur les paquets entrant de l'interface eth0 et c'est le genre d'instruction qui a été utilisé sur notre plateforme pour que la perturbation se fasse au niveau du testeur directement et l'exemple est montré dans la figure 55

```
#modprobe ifb
#ip link set dev ifb0 up
#tc qdisc add dev enp2s0 ingress
#tc filter add dev enp2s0 parent ffff:protocol ip u32 match u32 0 0 flowid 1:1
  action egress redirect dev ifb0
#tc qdisc add dev ifb0 root netem delay 100ms loss 1%
#tc class add dev enp2s0 parent 1:1 htb rate 100Mbps
```

Figure 55: Exemple commande Netem sur les paquets reçus.

IV.3 Présentation de la plateforme « QoEStreamingVideo » :

Au lancement de notre plateforme, la page d'authentification est la première page qui s'affiche pour accéder à l'espace de la plateforme, elle met en avant le logo choisi pour la plateforme QoEStreamingVideo et permet ainsi à un client ou à un administrateur d'accéder à son propre espace d'un côté ou à un client non enregistré de s'inscrire à la plateforme pour pouvoir utiliser ses fonctions (figure 56).



Figure 56: L'interface d'authentification.

IV.3.1 Espace administrateur

L'espace contient toutes les interfaces nécessaires pour que l'administrateur puisse exécuter les tâches administratives telles que la gestion des expériences, des vidéos, la configuration du réseau,

IV.3.1.1 Accueil administrateur

L'interface côté administrateur est assez simple et expose les différentes fonctions que l'administrateur peut exécuter. Un accueil avec un message de bienvenue ainsi que la date du jour (figure 57).



Figure 57: Accueil administrateur.

IV.3.1.2 Gestion de configuration de réseau local

La gestion du réseau local permet à l'administrateur d'ajouter et modifier un fichier de configuration auquel les tests y seront soumis (figure 58).

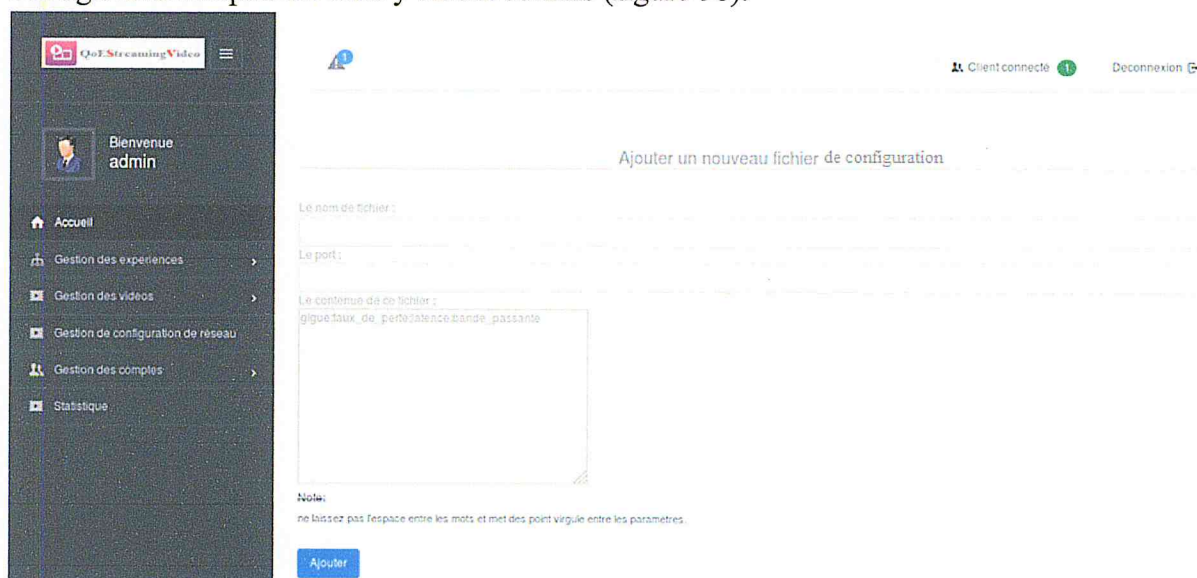


Figure 58: Gestion de configuration de réseau.

Le fichier de configuration est sous forme d'un fichier texte composé de ligne et de colonne. Chaque colonne représente un paramètre de QoS et la ligne représente la valeur des paramètres à exécuter simultanément au cours d'un teste.

IV.3.1.3 Gestion des expériences

L'onglet gestion des expériences est la fonction principale d'un administrateur, la fonction elle-même a deux fonctions secondaires : l'ajout d'une expérience et la recherche d'une expérience. La recherche ce fait pour modifier, masquer, supprimer une expérience ou bien pour rechercher un ou plusieurs tests de cette expérience. La figure 59 suivante montre c'est principales fonctionnalités.

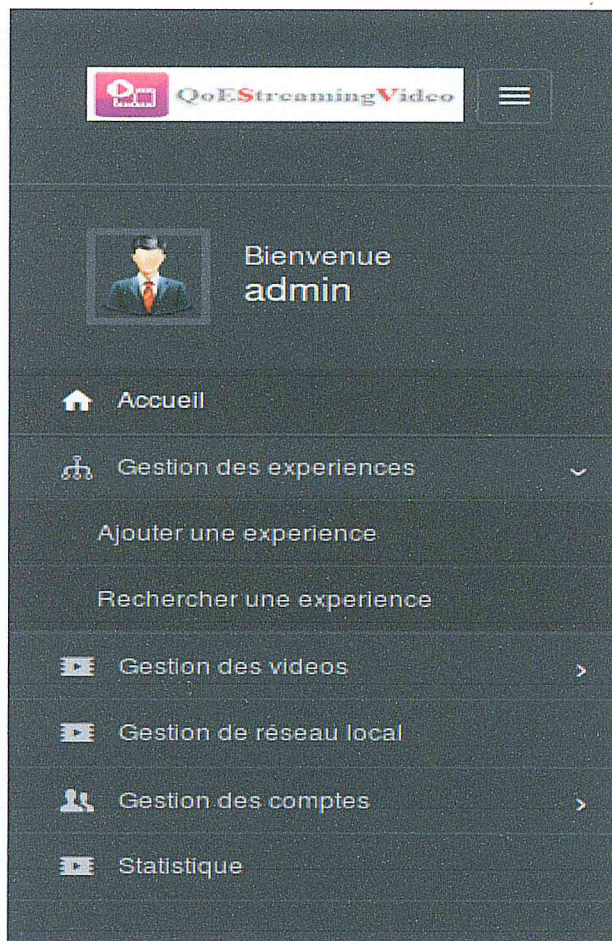


Figure 59: Gestion des expériences.

La fenêtre de la gestion des expériences est composée de deux parties. L'administrateur à la possibilité d'ajouter des tests avec le bouton « valider » après avoir rempli les champs (figure 60).

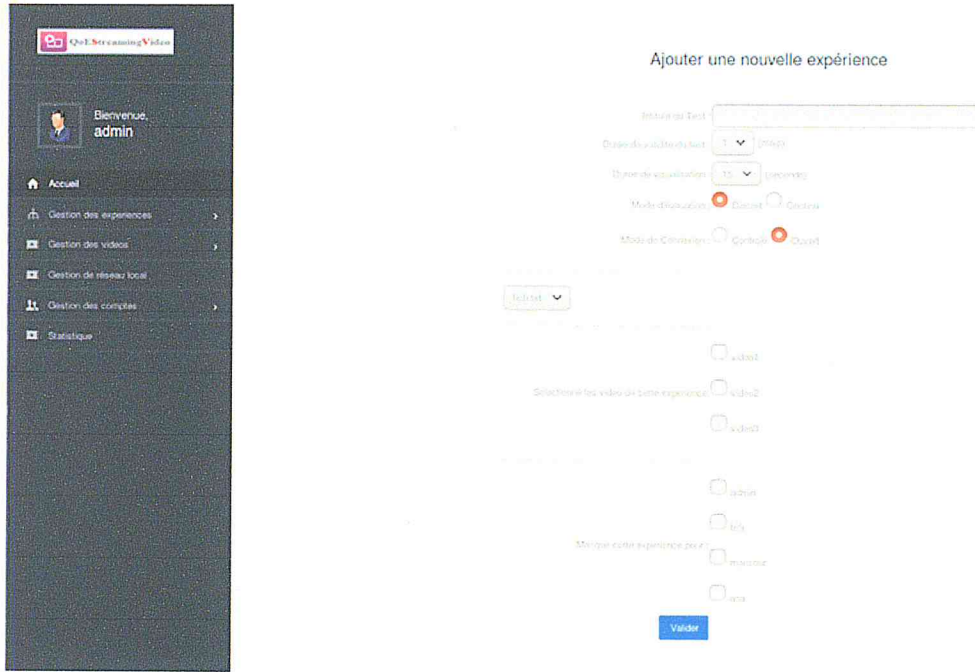


Figure 60: Ajout d'une nouvelle expérience.

La deuxième partie correspond à la recherche d'une expérience, après avoir consulté un test il peut le modifier ou le supprimer ou le masquer (figure 61).

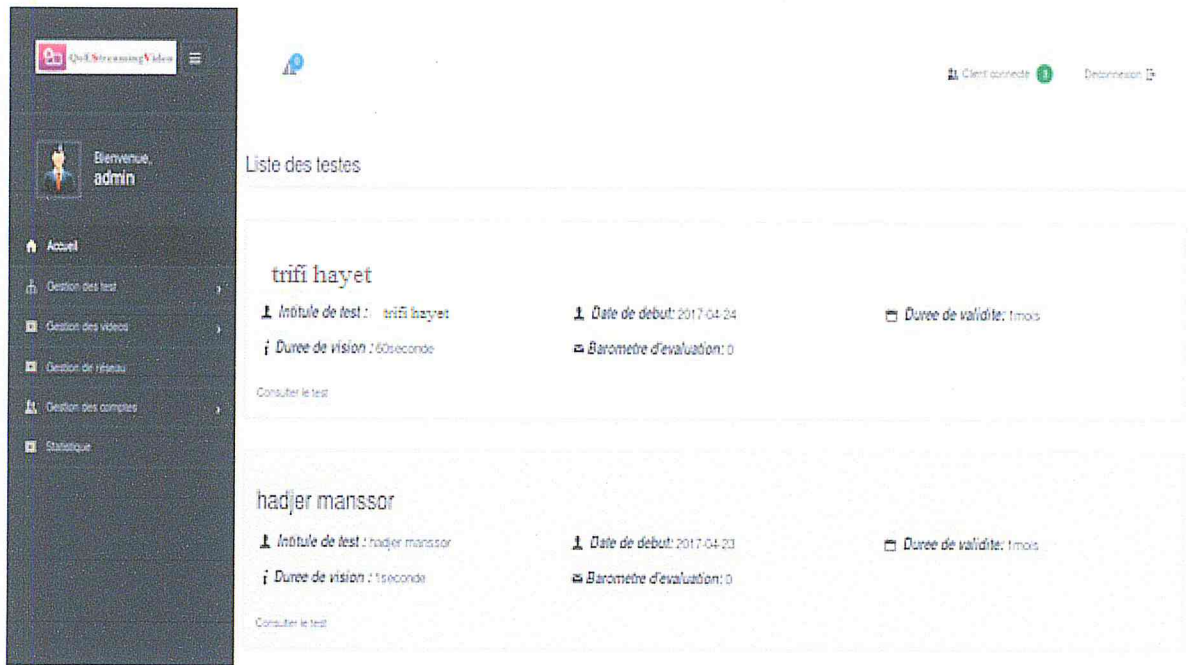


Figure 61: Liste des vidéos disponibles.

IV.3.2 Espace client

Le client pourra bénéficier de son propre espace. L'interaction de l'utilisateur avec la plateforme commence dès l'inscription (voir figure 62) jusqu'à l'achèvement des tests en passant par l'authentification bien évidemment, et voici la fonction de l'espace client :

IV.3.2.1 Inscription d'un utilisateur

La fonction d'inscription est la première étape pour un client dans le but d'utiliser la plateforme. La procédure d'inscription amène le client à fournir des informations personnels et qui seront propre à son compte tel que le nom d'utilisateur et le mot de passe (pour l'authentification), son nom et son prénom et sa date de naissance pour déterminer son âge étant donnée qu'une limite d'âge a été fixé et sa fréquence d'utilisation concernant l'utilisation du service de streaming vidéo.

The image shows a registration form on a dark blue background. At the top, there are two links: 'SE CONNECTER' and 'INSCRIPTION', with 'INSCRIPTION' being the active link. The form fields are as follows:

- Prénoms:** A text input field with a checkmark icon on the left and the placeholder text 'prénoms'.
- Nom d'utilisateur:** A text input field with a checkmark icon on the left and the placeholder text 'nom d'utilisateur'.
- Mot de passe:** A text input field with a key icon on the left and the placeholder text 'mot de passe'.
- Repeter le mot de passe:** A text input field with a key icon on the left and the placeholder text 'confirmer mot de passe'.
- Date de naissance:** Three dropdown menus for day, month, and year, with values '1', '1', and '1996' respectively.
- E-mail:** A text input field with an envelope icon on the left and the placeholder text 'E-mail'.
- Direz vous que vous etes:** A dropdown menu with the selected option 'Habitué au site de streaming'.
- Genre:** A dropdown menu with the selected option 'Homme'.

At the bottom of the form is a large white button labeled 'Inscrit'.

Figure 62: Formulaire d'inscription.

IV.3.2.2 Page d'accueil client

La page d'accueil client est relativement simple et expose toute les fonctions de l'espace client (figure 63).

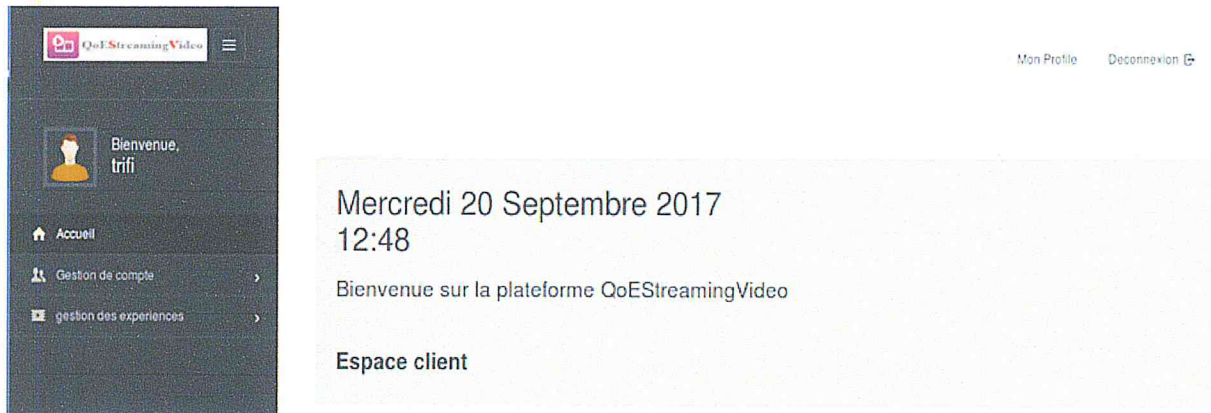


Figure 63: Accueil Client.

IV.3.2.3 Onglet test

L'onglet test présente à l'utilisateur les expériences qui sont disponible et configuré par l'administrateur (figure 64), le client n'a qu'à choisir une expérience à passer. Après avoir choisi une expérience, une liste des vidéos apparaitre (figure 65), le client doit choisir une vidéo et la lancer pour visionner (figure 66)



Figure 64: Liste des expériences disponibles.



Figure 65: Liste des vidéos disponibles d'une expérience choisie.

- Lancement d'un test

Une fois le test choisi, la vidéo se lance et l'utilisateur pourra visionner le clip de son choix, les paramètres seront appliqués au niveau du port client à partir du fichier de configuration sélectionné et le résultat obtenu est montré dans la figure 66.

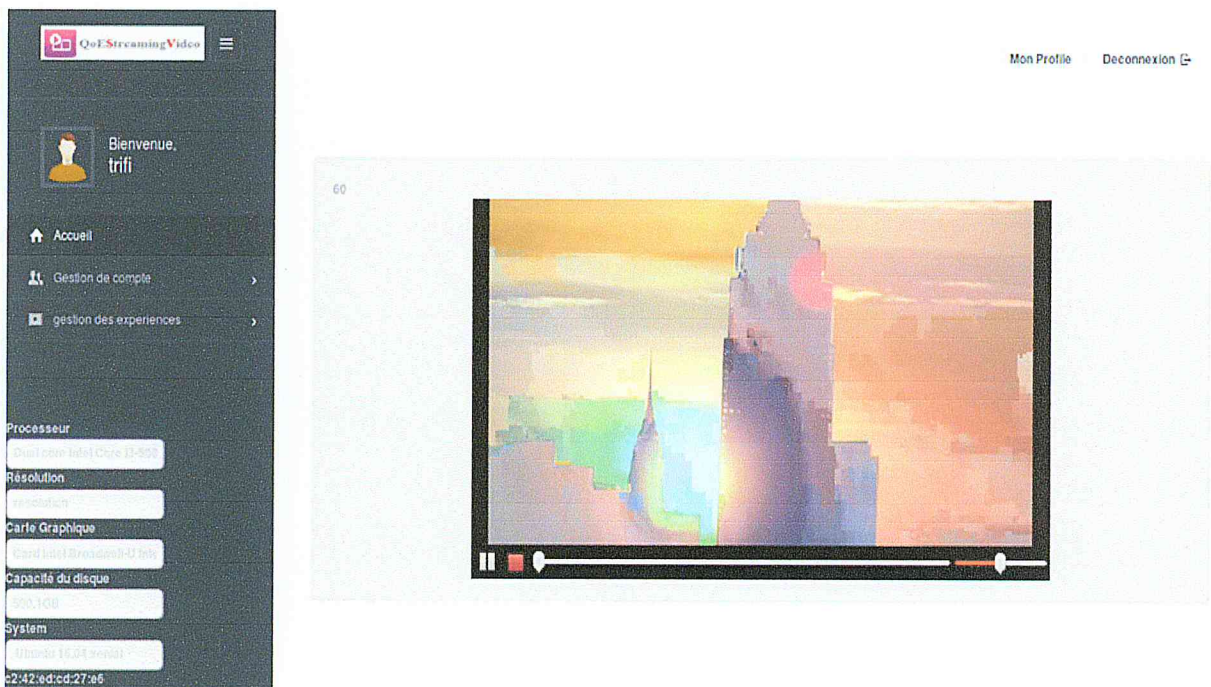


Figure 66: Lancement du test.

- Evaluation d'un teste

Une fois la durée du teste expiré, le client sera rediriger automatiquement vers la page d'évaluation où il pourra noter la vidéo visionnée (figure 67).

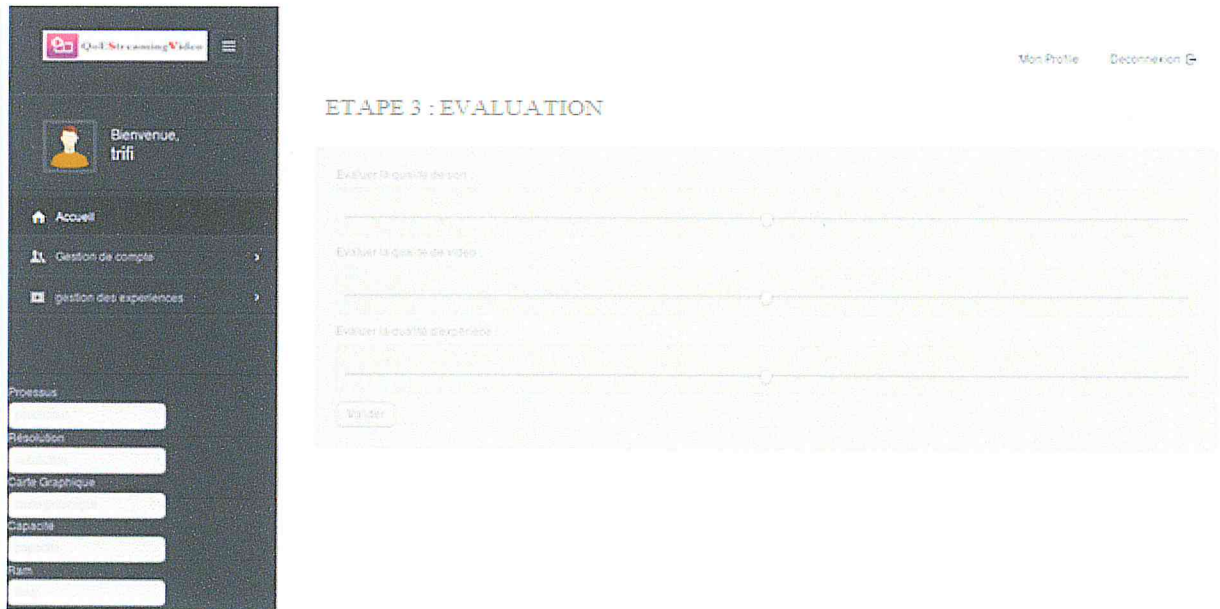


Figure 67: Evaluation de la QoE.

- **Résultats**

La figure 68 montre la partie Statistiques, l'administrateur a le choix d'établir les résultats en affichant les informations collectées tout au long d'une expérience. Si il veut afficher tous les paramètres collectés, l'administrateur doit coché les 5 cases sinon il coche la quelle correspondante au informations qu' il veut les afficher. La figure 69 montre un exemple de résultat en affichant les paramètres QoS et l'évaluation.

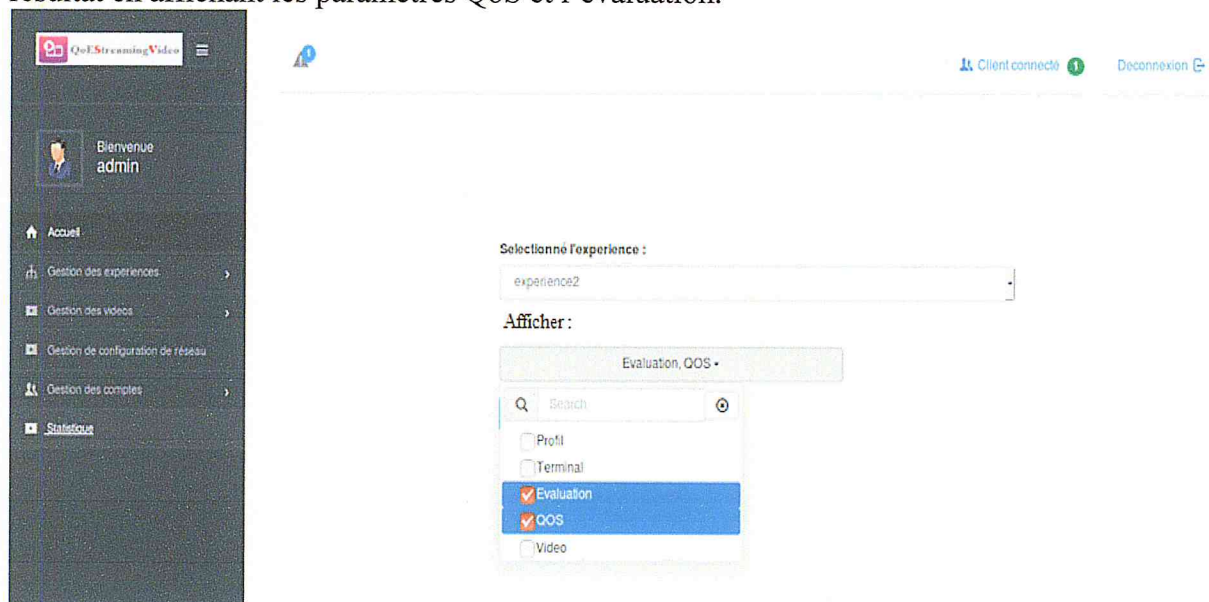


Figure 68 :Partie statistiques.



Figure 69: Résultat.

• Statistiques

Cette figure représente la fonctionnalité de l'extraction de donnée dans un fichier Excel par l'administrateur (figure 70).

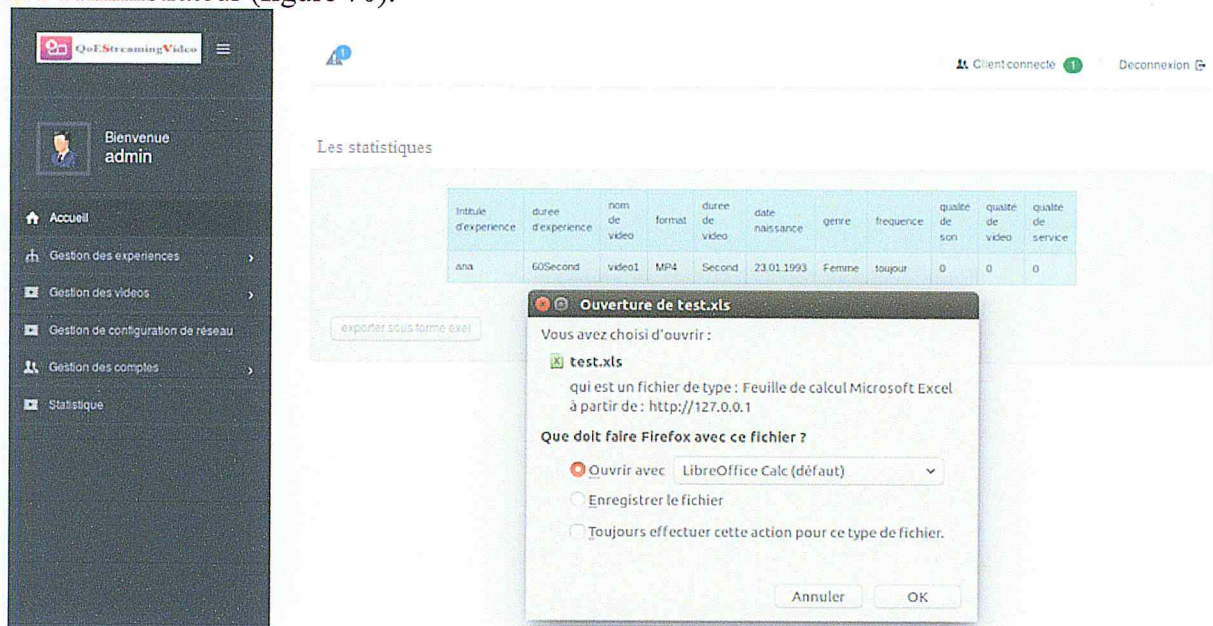


Figure 70: Extraction de donnée.

IV.4 Conclusion

Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre plateforme « QoEStreamingVideo », en spécifiant l'environnement de développement, l'implémentation de la base des données, la présentation des différentes interfaces qui compose les deux espaces développés (Administrateur et testeur) tout en exposant les fonctions qui constituent chaque espace.

En effet, nous avons achevé l'implémentation et les tests de tous les cas d'utilisation, tout en respectant la conception élaborée. En d'autres termes, nous détenons la version finale du logiciel, installée dans notre environnement de développement.

Conclusion générale

L'objectif visé à travers ce travail est de concevoir et d'implémenter une plateforme d'Expérimentation de la Qualité d'Expérience des utilisateurs de service de Streaming Vidéo.

Afin d'atteindre cet objectif, nous avons suivi plusieurs étapes. Dans un premier temps, nous avons commencé par une étude bibliographique sur la Qualité d'Expérience des services de Streaming vidéo, sur les principales métriques permettant d'évaluer le degré de satisfaction des utilisateurs des services vidéo, et sur les plateformes déjà réalisées dans la littérature qui nous ont bien aidés dans notre travail. Par la suite, nous avons fait une étude conceptuelle qui nous a permis de mettre en œuvre notre plateforme et de faciliter la collecte des informations des clients participants à l'expérience, des vidéos choisis pour les tests, de performance du réseau et des machines des clients. Enfin, nous avons réalisé notre plateforme, que nous avons appelé « QoEStreamingVideo » en utilisant les outils suivants : le serveur web LAMP, ffmpeg pour faire le streaming, vlc comme lecteur vidéo, NetEm pour émuler le réseau, inxi pour extraire les informations de la machine et mediainfo pour extraire les caractéristiques des vidéos.

Notre plateforme « QoEStreamingVideo » offre une facilité de manipulation ainsi qu'un système de collecte d'information automatisé ce qui est une innovation en quelque sorte. En effet, elle permet de collecter le profil de l'utilisateur (à l'inscription), machine (à la connexion via la commande inxi), vidéo (à la création d'expérience via l'outil mediainfo), QoS (à l'exécution de test via l'émulateur réseau Netem). A part le profil de l'utilisateur, les autres FI sont collectés de manière transparente par rapport au testeur. Par ailleurs, l'administrateur a le privilège de gérer les comptes des testeurs, d'ajouter les vidéos à la plateforme, de créer des fichiers de configuration du réseau local, de créer des expériences et de consulter les statistiques. Cette dernière fonctionnalité est destinée spécialement aux chercheurs du domaine pour exploiter les données collectées dans leurs études.

Chaque étude a ses avantages et ses limites. Ainsi, les principales améliorations qui peuvent être apportées sur notre plateforme sont :

- **Extension de l'environnement de test.** La plateforme, bien qu'elle soit implémentée seulement dans un environnement laboratoire où les paramètres réseau sont contrôlés, peut bien être extensible pour fonctionner dans un environnement réel sur internet.
- **Évaluation du multimédia.** La qualité du multimédia regroupe la qualité vidéo et de l'audio. Il serait intéressant d'étendre le travail pour évaluer le son et la vidéo ensemble en intégrant de nouveaux paramètres comme les caractéristiques du son.
- **Extension des paramètres réseau.** Les effets combinés d'autres facteurs peuvent être étudiés. Par exemple, le taux de perte était appliqué uniformément sur les paquets. Par ailleurs, il existe des moyens pour contrôler la distribution de ce taux, c'est ce qu'on appelle le taux de perte conditionnel. Il est important d'étudier l'impact de la distribution de la perte sur la qualité.

- **Extension du cadre des services.** Une fois la notion de QoE des services vidéo bien maîtrisée, nous envisageons de nous orienter vers les services multimédia où la synchronisation entre la vidéo et l'audio pose un grand défi.

Bibliographies

1. Recommandation UIT-T P.800. (1996). Méthodes d'évaluation objective et subjective de la qualité. 1-39. <https://www.itu.int/rec/T-REC-P.800-199608-I/fr>. consulté le 12 décembre 2016.
2. Brunnström, K. Beker, S.A. De Moor, K. Dooms, A. Egger, S. Garcia, M.N., & Zgank, A. (2013). Qualinet white paper on definitions of quality of experience. 1-15. <https://hal.archives-ouvertes.fr/hal-00977812/document>. Consulté le 25 décembre 2016.
3. Leroy, L. Lhote, B. (2002). Téléchargement Streaming. 1-13. https://wapiti.telecom-lille.fr/commun/ens/peda/options/st/rio/pub/exposes/exposesrio2002/Leroy-Lhote/Telechargement_Streaming.pdf. Consulté le 8 janvier 2017.
4. Burk, T. (2004). Techniques de diffusion vidéo sur l'Internet.1-9. <http://cpham.perso.univ-pau.fr/MIM2/BIBLIO/03-04/diff-video-rapport.pdf>. Consulté de 8 janvier 2017.
5. Vidéo à la demande. (s. d.). Dans *Wikipédia, l'encyclopédie libre*. Repéré le 9 janvier 2017 à https://fr.wikipedia.org/wiki/Vid%C3%A9o_%C3%A0_la_demande.
6. Streaming. (s. d.). Dans *Wikipédia, l'encyclopédie libre*. Repéré le 10 janvier 2017 à <https://fr.wikipedia.org/wiki/Streaming>.
7. Ben Ameer, C. (2015). TCP Protocol Optimization for HTTP Adaptive Streaming. En ligne. 1-199. <https://hal.archives-ouvertes.fr/tel-01249840/document>. Consulté le 15 janvier 2017.
8. Real-time Transport Control Protocol. (s. d.). Dans *Wikipédia, l'encyclopédie libre*. Repéré le 28 janvier 2017 à https://fr.wikipedia.org/wiki/Real-time_Transport_Control_Protocol.
9. Aroussi, S. (2011). Méta-Routage basé sur la Qualité d'Expérience dans le cadre des Applications Vidéo. (Mémoire de Magistère, Ecole nationale Supérieure d'Informatique, Alger).
10. Youssari, H., Zahraoui, N. (2014). Conception et Implémentation d'une Plateforme d'Expérimentation pour Évaluer la Qualité d'Expérience des Utilisateurs des Services de Streaming vidéo. (Mémoire de master, université Saad Dahleb, Blida).
11. Recommandation ITU-T. G.1080. (2008). Quality of experience Requirements for IPTV Services. 1-44. <https://www.itu.int/rec/T-REC-P.800-199608-I/fr>. consulté le 29 janvier 2017.
12. Rahrer, T. Fiandra, R. et Wright, S. (2006). Triple-play Services Quality of Experience (QoE) Requirements, Architecture & Transport Working Group, DSL Forum, Technical Report (TR-126). 1-129. <https://www.broadband-forum.org/technical/download/TR-126.pdf>. Consulté le 1 février 2017.
13. De Marez, L. De Moor, K. (2007). The challenge of user- and QoE-centric research and product development in today's ICT environment. *Observatorio (OBS*)*, 1(3), 1-22.

14. Mamadou Tourad, D. (2015). Quality of experience and video services adaptation. Networking and Internet Architecture [cs.NI].1-139.
15. Chen, C. Liu, Q. (2015). QoE in Video Transmission: A user Experience Driven Strategy. 1-19. <https://www.researchgate.net/publication/309323744>. Consulté le 8 février 2017.
16. Mushtaq, M. S. (2016). Contribution of Quality of Experience to optimize multimedia services: the case study of video streaming and VoIP. Computer science.1-188. <https://tel.archives-ouvertes.fr/tel-01281367/document>. Consulté le 12 février 2017.
17. Amour, L. Souihi, S. Hoceini, S. Melouk, A. (2015a). An open source Platform for perceived video quality evaluation. 1-3. https://www.researchgate.net/publication/301464319_An_Open_Source_Platform_for_Perceived_Video_Quality_Evaluation. Consulté le 12 février 2017.
18. Amour, L. Souihi, S. Hoceini, S. Melouk, A. (2015b). A Hierarchical Classification Model of QoE. 1-14.
19. Aroussi, S. Mellouk, A. (2014). Survey on Machine Learning-based QoE-QoS Correlation Models. 1-5.
20. la plateforme (informatique).le parisien. [http://dictionnaire.sensagent.leparisien.fr/Plateforme%20\(informatique\)/fr-fr/](http://dictionnaire.sensagent.leparisien.fr/Plateforme%20(informatique)/fr-fr/). consulté le 10 février 2017.
21. Shahid, M. Søgaard, J. Pokhrel, J. Brunnström, K. Wang, K. Tavakol, S. and Garcia, N. (2014). Crowdsourcing based subjective quality assessment of adaptive video streaming.
22. Yen, Y. C. Chu, C. Y. Yeh, S. L. Chu, H. H. and Huang, P. (2013). Lab Experiment vs. Crowdsourcing: A Comparative User Study on Skype Call Quality.1-15. <https://pdfs.semanticscholar.org/2813/088cf41b36bbfca702ea46b91e2581c7c01e.pdf> . Consulté le 12 février 2017.
23. Pokhrel, J. (2014). Analyse intelligente de la qualité d'expérience (QoE) dans les réseaux de diffusion de contenu Web et Multimédia. (thèse de doctorat, Université Evry Val d'Essonne, sud paris).
24. Hobfeld, T. Hirth, M. Judith, R. Mazza, F. Korshunov, P. Naderi, B., Egger, S. (2014). Best Practices and Recommendations for Crowdsourced QoE- Lessons learned from the Qualinet Task Force. 1-29.
25. Laghari, K. U. R. (2012). On quality of experience (QoE) for multimedia services in communication ecosystem. (thèse de doctorat, Université Pierre et Marie Curie, sud paris). (2012TELE0008).
26. Recommendation ITU-T P.910. (2008). Subjective vidéo quality assessment methods for multimedia applications.1-42.
27. Benyettou, L. (2012). Mise en œuvre d'un modèle de corrélation QoS-QoE. Réseaux et télécommunications [cs.NI].1-44. <https://dumas.ccsd.cnrs.fr/dumas-00725182/document>. Consulté le 10 mars 2017.
28. P, Gérard. (2008). «Processus de développement logiciel », Université de Paris 13. Repéré le 30 Juin 2017 à <https://cyclededeveloppementdunlogiciel.wordpress.com/le-modele-en-cascade/>
29. J.Steffe. (2005). « Cours UML», ENITA de Bordeaux.
30. Muller. (2002). « Modélisation objet avec UML », EYROLLES.
31. Bres. (1993). « Atelier de génie logiciel », Masson.

32. Diagramme des cas d'utilisation. Repéré le 30 mars 2017 à <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/use-case-diagramme>
33. Diagramme de séquence. Repéré le 30 Juin 2017 à <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/sequence-diagram>
34. Diagramme de classes. Repéré le 30 Juin 2017 à <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/diagramme-de-classe>
35. Ubuntu. Repéré à <https://ubuntu-fr.org/>. Consulté le 17 mars 2017.
36. Apache. Repéré à <https://httpd.apache.org/>. Consulté le 17 mars 2017.
37. MySQL. Repéré à <https://www.mysql.com/fr/>. Consulté le 17 mars 2017.
38. HTML. Repéré à <http://www.w3.org/html/> . Consulté le 5 Juillet 2017.
39. CSS. Repéré le 5 Avril 2017 à <https://developer.mozilla.org/fr/docs/Web/CSS> .
40. FFmpeg. Repéré à <https://www.ffmpeg.org/about.html>. Consulté le 10 Juin 2017.
41. FFserver. Repéré à <https://trac.ffmpeg.org/wiki/ffserver>. Consulté le 11 Juin 2017.
42. VLC. doc/streaming-howto/fr/ch05. www.videolan.org. Consulté le 12 Juin 2017.
43. Mediainfo. <http://mediaarea.net/fr/MediaInfo>. Consulté le 14 Juin 2017
44. INXI. Repéré à <http://la-vache-libre.org/obtenez-un-grand-nombre-dinformations-sur-votre-systeme-grace-a-inxi/>. Consulté le 10 septembre 2017.
45. Netem Linux Foundation. <http://www.linuxfoundation.org>. [En ligne] <http://www.linuxfoundation.org/collaborate/workgroups/networking/netem> .

Bibliographies

1. Recommandation UIT-T P.800. (1996). Méthodes d'évaluation objective et subjective de la qualité. 1-39. <https://www.itu.int/rec/T-REC-P.800-199608-I/fr>. consulté le 12 décembre 2016.
2. Brunnström, K. Beker, S.A. De Moor, K. Dooms, A. Egger, S. Garcia, M.N., & Zgank, A. (2013). Qualinet white paper on definitions of quality of experience. 1-15. <https://hal.archives-ouvertes.fr/hal-00977812/document>. Consulté le 25 décembre 2016.
3. Leroy, L. Lhote, B. (2002). Téléchargement Streaming. 1-13. https://wapiti.telecom-lille.fr/commun/ens/peda/options/st/rio/pub/exposes/exposesrio2002/Leroy-Lhote/Telechargement_Streaming.pdf. Consulté le 8 janvier 2017.
4. Burk, T. (2004). Techniques de diffusion vidéo sur l'Internet.1-9. <http://cpham.perso.univ-pau.fr/MIM2/BIBLIO/03-04/diff-video-rapport.pdf>. Consulté de 8 janvier 2017.
5. Vidéo à la demande. (s. d.). Dans *Wikipédia, l'encyclopédie libre*. Repéré le 9 janvier 2017 à https://fr.wikipedia.org/wiki/Vid%C3%A9o_%C3%A0_la_demande.
6. Streaming. (s. d.). Dans *Wikipédia, l'encyclopédie libre*. Repéré le 10 janvier 2017 à <https://fr.wikipedia.org/wiki/Streaming>.
7. Ben Ameer, C. (2015). TCP Protocol Optimization for HTTP Adaptive Streaming. En ligne. 1-199. <https://hal.archives-ouvertes.fr/tel-01249840/document>. Consulté le 15 janvier 2017.
8. Real-time Transport Control Protocol. (s. d.). Dans *Wikipédia, l'encyclopédie libre*. Repéré le 28 janvier 2017 à https://fr.wikipedia.org/wiki/Real-time_Transport_Control_Protocol.
9. Aroussi, S. (2011). Méta-Routage basé sur la Qualité d'Expérience dans le cadre des Applications Vidéo. (Mémoire de Magistère, Ecole nationale Supérieure d'Informatique, Alger).
10. Youssari, H., Zahraoui, N. (2014). Conception et Implémentation d'une Plateforme d'Expérimentation pour Évaluer la Qualité d'Expérience des Utilisateurs des Services de Streaming vidéo. (Mémoire de master, université Saad Dahleb, Blida).
11. Recommendation ITU-T. G.1080. (2008). Quality of experience Requirements for IPTV Services. 1-44. <https://www.itu.int/rec/T-REC-P.800-199608-I/fr>. consulté le 29 janvier 2017.
12. Rahrer, T. Fiandra, R. et Wright, S. (2006). Triple-play Services Quality of Experience (QoE) Requirements, Architecture & Transport Working Group, DSL Forum, Technical Report (TR-126). 1-129. <https://www.broadband-forum.org/technical/download/TR-126.pdf>. Consulté le 1 février 2017.
13. De Marez, L. De Moor, K. (2007). The challenge of user- and QoE-centric research and product development in today's ICT environment. Observatorio (OBS*), 1(3), 1–22.

14. Mamadou Tourad, D. (2015). Quality of experience and video services adaptation. Networking and Internet Architecture [cs.NI].1-139.
15. Chen, C. Liu, Q. (2015). QoE in Video Transmission: A user Experience Driven Strategy. 1-19. <https://www.researchgate.net/publication/309323744>. Consulté le 8 février 2017.
16. Mushtaq, M. S. (2016). Contribution of Quality of Experience to optimize multimedia services: the case study of video streaming and VoIP. Computer science.1-188. <https://tel.archives-ouvertes.fr/tel-01281367/document>. Consulté le 12 février 2017.
17. Amour, L. Souihi, S. Hoceini, S. Melouk, A. (2015a). An open source Platform for perceived video quality evaluation. 1-3. https://www.researchgate.net/publication/301464319_An_Open_Source_Platform_for_Perceived_Video_Quality_Evaluation. Consulté le 12 février 2017.
18. Amour, L. Souihi, S. Hoceini, S. Melouk, A. (2015b). A Hierarchical Classification Model of QoE. 1-14.
19. Aroussi, S. Mellouk, A. (2014). Survey on Machine Learning-based QoE-QoS Correlation Models. 1-5.
20. la plateforme (informatique).le parisien. [http://dictionnaire.sensagent.leparisien.fr/Plateforme%20\(informatique\)/fr-fr/](http://dictionnaire.sensagent.leparisien.fr/Plateforme%20(informatique)/fr-fr/). consulté le 10 février 2017.
21. Shahid, M. Sogaard, J. Pokhrel, J. Brunnström, K. Wang, K. Tavakol, S. and Garcia, N. (2014). Crowdsourcing based subjective quality assessment of adaptive video streaming.
22. Yen, Y. C. Chu, C. Y. Yeh, S. L. Chu, H. H. and Huang, P. (2013). Lab Experiment vs. Crowdsourcing: A Comparative User Study on Skype Call Quality.1-15. <https://pdfs.semanticscholar.org/2813/088cf41b36bbfca702ea46b91e2581c7c01e.pdf> . Consulté le 12 février 2017.
23. Pokhrel, J. (2014). Analyse intelligente de la qualité d'expérience (QoE) dans les réseaux de diffusion de contenu Web et Multimédia. (thèse de doctorat, Université Evry Val d'Essonne, sud paris).
24. Hobfeld, T. Hirth, M. Judith, R. Mazza, F. Korshunov, P. Naderi, B., Egger, S. (2014). Best Practices and Recommendations for Crowdsourced QoE- Lessons learned from the Qualinet Task Force. 1-29.
25. Laghari, K. U. R. (2012). On quality of experience (QoE) for multimedia services in communication ecosystem. (thèse de doctorat, Université Pierre et Marie Curie, sud paris). (2012TELE0008).
26. Recommendation ITU-T P.910. (2008). Subjective video quality assessment methods for multimedia applications.1-42.
27. Benyettou, L. (2012). Mise en œuvre d'un modèle de corrélation QoS-QoE. Réseaux et télécommunications [cs.NI].1-44. <https://dumas.ccsd.cnrs.fr/dumas-00725182/document>. Consulté le 10 mars 2017.
28. P, Gérard. (2008). «Processus de développement logiciel », Université de Paris 13. Repéré le 30 Juin 2017 à <https://cyclededeveloppementdunlogiciel.wordpress.com/le-modele-en-cascade/>
29. J.Steffe. (2005). « Cours UML», ENITA de Bordeaux.
30. Muller. (2002). « Modélisation objet avec UML », EYROLLES.
31. Bres. (1993). « Atelier de génie logiciel », Masson.

32. Diagramme des cas d'utilisation. Repéré le 30 mars 2017 à <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/use-case-diagramme>
33. Diagramme de séquence. Repéré le 30 Juin 2017 à <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/sequence-diagram>
34. Diagramme de classes. Repéré le 30 Juin 2017 à <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/diagramme-de-classe>
35. Ubuntu. Repéré à <https://ubuntu-fr.org/>. Consulté le 17 mars 2017.
36. Apache. Repéré à <https://httpd.apache.org/>. Consulté le 17 mars 2017.
37. MySQL. Repéré à <https://www.mysql.com/fr/>. Consulté le 17 mars 2017.
38. HTML. Repéré à <http://www.w3.org/html/> . Consulté le 5 Juillet 2017.
39. CSS. Repéré le 5 Avril 2017 à <https://developer.mozilla.org/fr/docs/Web/CSS> .
40. FFmpeg. Repéré à <https://www.ffmpeg.org/about.html>. Consulté le 10 Juin 2017.
41. FFserver. Repéré à <https://trac.ffmpeg.org/wiki/ffserver>. Consulté le 11 Juin 2017.
42. VLC. doc/streaming-howto/fr/ch05. www.videolan.org. Consulté le 12 Juin 2017.
43. Mediainfo. <http://mediaarea.net/fr/MediaInfo>. Consulté le 14 Juin 2017
44. INXI. Repéré à <http://la-vache-libre.org/obtenez-un-grand-nombre-dinformations-sur-votre-systeme-grace-a-inxi/>. Consulté le 10 septembre 2017.
45. Netem Linux Foundation. <http://www.linuxfoundation.org>. [En ligne] <http://www.linuxfoundation.org/collaborate/workgroups/networking/netem> .

