

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE D'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE**

ECOLE NATIONALE POLYTECHNIQUE



**DEPARTEMENT D'ELECTRONIQUE
OPTION: SIGNAL & COMMUNICATIONS**

MEMOIRE DE MAGISTER EN ELECTRONIQUE

Présenté par :

M. KADIK Rachid

*Ingénieur en Télécommunications, l'institut National des Télécommunications
et de la Technologie de l'Information et de la Communication (INT-TIC)*

**MISE EN ŒUVRE DE LA NORME DIFFSERV SOUS
NS2 POUR LA TRANSMISSION DE VIDEO**

Membre de jury:

Présidente

Mme L.HAMMAMI

Professeur ENP

Rapporteur

M. BERKANI Daoud

Professeur ENP

Examineur

Mlle M.GUERTI

Professeur ENP

M.R.SADOUNE

C.C.ENSP

M.R.ZERGUI

C.C.ENSP

Decembre 2009

DEDECAGE

ملخص: تعد الإنترنت تاريخياً مصممة على عرض الخدمات التي تسمى بـ: Best-Effort ، حيث أن لجميع الحزم نفس الأولوية عند عبور الشبكة. غير أن التطبيقات التي يحتاج إرسالها إلى وقت حقيقي، مدة الإرسال، الضياع لديهم قيم محدودة وإلا كانت المعلومات لا دلالة لها. نوعية الخدمة هي التي تسمح بضمان سير المعلومات. إجابة على طلب تحسين نوعية الخدمة، IETF عرفت نظامين DiffServ و IntServ. هذا الأخير يعتمد على آلية حجز الوسائل، الثاني DiffServ يضع علامات على كل الحزم في حافة الشبكة مع عدد محدود من أقسام التوجيه. من هذه الأقسام، التوجيه المضمون AF الذي يوجه الحزم إلى أربعة أقسام مع ثلاث أولويات لطرح الحزم في كل قسم. يهدف هذا العمل إلى التفريق بين الصور فيديو مضغوط. مثلاً فيديو مضغوط بـ: MPEG لديه ثلاث الصور I, B, و P. الصور I لها أهمية أكثر الصور P كما له أهمية أكثر من B، فنهدف إلى توفيق بين أقسام الخدمات و أنواع الصور.

المفاتيح: الشبكات الحاسوبية، الضغط MPEG، DiffServ، NS.

Résumé : Historiquement, L'Internet était conçu pour offrir un service dit Best-Effort, où tous les paquets ont une priorité équitable à travers le réseau. Pour des applications temps réel, le délai, la gigue et les pertes ont une valeur bornée au sens où une fois cette valeur dépassée, les données n'ont plus de signification. La qualité de service est la méthode permettant de garantir le trafic de données. En réponse à cette demande d'amélioration de la qualité de service, l'IETF a défini deux modèles : DiffServ et IntServ. Ce dernier repose sur un mécanisme de réservation des ressources, le second modèle DiffServ, consiste à marquer les paquets individuels dans les bordures du réseau avec un nombre réduit de classes d'acheminement. L'une de ces classes « Acheminement Assuré » (AF : Assured Forwarding) définit l'acheminement des paquets IP dans quatre classes avec trois priorités de rejet dans chaque classe. Dans ce travail, nous allons faire une différenciation entre les trames du flux compressé générées par les codecs vidéos, Par exemple, au niveau des trames, une vidéo compressée avec MPEG contient trois types de trames I, P et B. La trame I est plus importante qu'une trame P qui est plus importante qu'une trame B. Le but de ce travail est la mise en correspondance des classes de services avec les différents types de trame.

Mots clefs : Réseau informatique, compression MPEG, DiffServ, NS.

Abstract: Historically, the Internet was designed to offer a service called Best-Effort, where all the packets have an equitable priority over the network. For real-time applications, the delay, jitter, and losses packets have limited values, when one of this values exceed; the data doesn't have any signification. Quality of service is the method to ensure the traffic data. As answer to this request for improvement of quality of service, two models: IntServ and DiffServ have been defined by IETF. The first one is based on a resources reservation, the second marks individual packets in the network edge with a reduced number of routing classes. One of these classes AF: Assured Forwarding which defines the routing of packets IP in four classes with three priorities of rejection in each class. In this work, we will make a differentiation between the frames of the compressed flows generated by the video codec, For example, the MPEG coding video contains three types of frames I, P and B. frame I is more important than P which is more important than B. the aim of this work is mapping of these classes of service with the various types of frames.

Key word: Networking, MPEG compression, DiffServ, NS.

Ces dernières années, les évolutions conjointes de l'informatique et des télécommunications ont conduit à l'émergence de nouveaux types d'applications distribuées contraintes, telles que les applications multimédias (visioconférences, vidéo à la demande, ...) ou la simulation interactive distribuée.

Contrairement aux applications classiques, du type transfert de fichiers, consultation de pages du Web ou encore messagerie électronique, ces applications présentent des exigences nouvelles en terme de communication : délai de transit borné, fiabilité non plus totale mais partielle, garantie de bande passante ..., auxquelles ont à faire face tant les réseaux physiques qui les supportent, que l'Internet, la solution incontournable d'interconnexion de ces réseaux.

Dans ce contexte et en raison des limites conceptuelles de l'Internet (initialement conçu pour assurer le transport des applications classiques), résulte la problématique très générale consistant à faire en sorte que ces nouvelles applications puissent être distribuées en satisfaisant leurs contraintes entre deux ou plusieurs hôtes de l'Internet, indépendamment de leur point d'accès (fixe ou mobile).

Dans cette optique, de nombreux travaux de recherche se sont ainsi focalisés depuis dix ans sur cette problématique suivant deux approches complémentaires :

- La première approche préconise d'accroître la complexité des applications pour adapter les logiciels d'application aux variations de performance (gigue et/ou pertes) de l'Internet. Initiée par (Bolot, 1994), cette approche est basée sur l'utilisation des protocoles RTP/RTCP (Schulzrinne, 1999) et est implémentée dans plusieurs applications audio et vidéo de l'Internet actuel ;
- la seconde approche préconise de revoir les niveaux Transport et IP de l'architecture de l'Internet afin d'offrir aux applications des garanties de « Qualité de Service » (QoS). Suivant cette approche, différents travaux ont été initiés et sont encore menés, tant au niveau IP (nouveaux modèle de services, nouvelles architectures - IntServ, DiffServ, ...) qu'au niveau Transport (amélioration des mécanismes existants, extension des fonctionnalités, nouveaux protocoles, ...).

TABLE DE MATIERE

CHAPITRE I.....	8
NOTIONS DE BASE SUR L'IMAGE ET LA COMPRESSION NUMERIQUE. 8	
I.1. Introduction :.....	3
I.2. Notions de base sur l'image numérique :.....	3
I.2.1 Définition de l'image :.....	3
I.2.2 Images numériques :.....	3
I.2.3 Caractéristiques d'une image numérique :.....	4
I.3. But de la compression numérique :.....	6
I.3.1 Principe de la compression de la vidéo.....	7
I.3.2 Types de Compressions vidéo:	8
I.3.3 Les redondances de l'image vidéo :.....	8
I.3.4 Quelques ordres de grandeurs :.....	9
I.4. Mesure de la qualité vidéo.....	17
I.4.1 Evaluation subjective :.....	17
I.4.2 Evaluation objective :.....	17
I.4.3 Applications de la vidéo sur un Réseau IP :.....	18
CHAPITRE II :	20
LES Réseaux Informatiques	20
I.1. Introduction :.....	21
I.2. Le modèle OSI.....	21
I.3. Le modèle TCP/IP :.....	22
I.4. Les principaux protocoles et applications de l'environnement TCP/IP :.....	24
I.5. Le protocole TCP :	25
I.6. Le protocole UDP	28
I.7. Le protocole IP.....	28
I.8. Protocoles de transmission Temps réel :.....	Erreur ! Signet non défini.

CHAPITRE III.....	34
QUALITE DE SERVICE DANS UN RESEAU IP	34
III.1. La Qualité de service dans les réseaux IP	35
III.1.1. Importance de la qualité de service :.....	35
III.1.2. Critères de QoS dans l'Internet :	36
III.1. Les mécanismes de QoS sur IP :.....	38
III.1.1. Integrated Services et signalisation RSVP :	38
III.1.2. Le modèle DiffServ.....	39
III.1.3. Les Comportements Standardisés.....	47
III.2. La diffusion consciente de DiffServ.....	49
III.2.1. Concept fondamental.....	49
CHAPITRE IV	52
APPLICATIONS ET RESULTATS PRATIQUES.....	52
IV.1. Simulation.....	53
IV.2. Présentation de NS-2 :.....	53
IV.2.1. Développement du simulateur NS.....	53
IV.2.2. Le simulateur NS-2.....	53
IV.2.3. Développement de nouveaux composants.....	54
IV.2.4. Architecture du réseau.....	55
IV.2.5. 5. Interprétation des résultats	57
IV.3. Architecture de la Simulation:	60
IV.3.1. Outils développés :	61
IV.3.2. Scénarios simulés :	61
IV.3.3. Etapes de la simulation :	61
➔ Encodage de la vidéo :	62
➔ Packetization et détection des trames	63
➔ Simulation avec NS-2 :	64
➔ Génération de la vidéo reçue	66
➔ Décodage de la vidéo reçue	66
➔ Correction de la vidéo reçue décodée :	66
➔ Calcul du PSNR.....	67
➔ Calcule de MOS.....	67
IV.4. Résultats de la simulation	67
IV.5. Conclusion	74
Bibliographie :.....	75

CHAPITRE I

NOTIONS DE BASE SUR L'IMAGE ET LA COMPRESSION NUMERIQUE

1.1. Introduction :

Chaque séquence vidéo est une combinaison de plusieurs trames (frames) (image), notre travail est une application sur les séquences vidéo, donc sur les images Composant la vidéo.

Commençons par un petit rappel et des notions de base sur l'image numérique, ce chapitre donne toutes les clés permettant de comprendre les principes de la compression numérique appliquée à la vidéo, technique complexe qui, en s'imposant chaque jour davantage, a quasiment banalisé la manipulation des images vidéo numériques dans les différents domaines, la vidéo conférence, les multimédias, la téléphonie visuelle,...

Nous décrivons donc les normes de codage des images JPEG, MPEG et leur application dans le domaine du réseau.

1.2. Notions de base sur l'image numérique :

1.2.1 Définition de l'image :

L'image est une représentation d'un objet par la peinture, la sculpture, le dessin, la photographie, le film. C'est aussi un ensemble structuré d'informations qui, après affichage sur écran, ont une signification pour l'œil humain. Elle peut être décrite sous la forme d'une fonction $I(x, y)$ de brillance analogique continue, définie dans le domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur. Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation. [GMC94]

1.2.2 Images numériques :

Contrairement aux images obtenues à l'aide d'un appareil photo, ou dessinées sur du papier, les images manipulées par un ordinateur sont numériques (représentées par une série de bits). L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter. La numérisation d'une image est la conversion de celle-ci de son état analogique en une image numérique représentée par une matrice bidimensionnelles de valeurs numériques $f(x, y)$ où : x, y : coordonnées cartésiennes d'un point de l'image. $f(x, y)$ représente niveau de gris en ce point. Pour des raisons de commodité de représentation pour l'affichage et l'adressage, les données images sont généralement rangées sous formes de tableau I de n lignes et p colonnes. Chaque élément $I(x, y)$ représente un pixel de l'image et à sa valeur est associé un niveau de gris codé sur m bits (2^m niveaux de gris ; $0 = \text{noir}$; $2^m - 1 = \text{blanc}$). La valeur en

chaque point exprime la mesure d'intensité lumineuse perçue par le capteur. [GM93], [site4].

Il existe deux sortes d'images numériques : les images matricielles et les images vectorielles. Dans une image vectorielle les données sont représentées par des formes géométriques simples qui sont décrites d'un point de vue mathématique. Par exemple, un cercle est décrit par une information de type (cercle, position du centre, rayon). Ces images sont essentiellement utilisées pour réaliser des schémas ou des plans. Ces images présentent deux avantages : elles occupent peu d'espace en mémoire et peuvent être redimensionnées sans perte d'informations. Une image matricielle est formée d'un tableau de points ou de pixels. Plus la densité des points est élevée, plus le nombre d'informations est grand et plus la résolution de l'image est élevée. Corrélativement, l'espace occupé en mémoire et la durée de traitement seront d'autant plus grands.

1.2.3 Caractéristiques d'une image numérique :

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants:

➤ **Pixel :**

Contraction de l'expression anglaise " Picture Elements ": éléments d'image, le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression. La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la Figure ci-dessous :

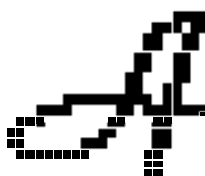


Figure I.1 : Architecture des pixels de la lettre A.

La quantité d'information que véhicule chaque pixel donne des nuances entre images monochromes et images couleurs. Dans le cas d'une image monochrome, chaque pixel est codé sur un bit, et la taille mémoire nécessaire pour afficher une telle image est directement liée à la taille de l'image. Dans une image couleur (R.V.B.), un pixel peut être représenté sur trois octets : un octet pour chacune des couleurs : rouge (R), vert (V) et bleu (B).

➤ **Dimension :**

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

➤ **Résolution :**

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur. Plus ce nombre est grand plus la résolution est meilleure.

➤ **Bruit :**

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins. Il provient de l'éclairage des dispositifs optiques et électroniques du capteur.

➤ **Contours et textures :**

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes.

➤ **Luminance :**

La luminance est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface. Pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes);
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

➤ **Contraste :**

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport :

$$C = \frac{L_1 - L_2}{L_1 + L_2} \dots\dots\dots(I.1)$$

➤ **Image à niveaux de gris :**

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255.

➤ **Image en couleurs :**

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs,...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.).[site4], [GMC94].

1.3. But de la compression numérique :

Si la compression numérique fait beaucoup parler d'elle depuis quelques années, il ne faut pas oublier que l'idée de réduire la quantité d'information du signal vidéo n'est guère nouvelle. Depuis les débuts de l'histoire de la télévision, on a toujours cherché à exploiter les caractéristiques psycho-visuelles de l'œil humain pour restreindre à des valeurs raisonnables, l'encombrement du signal vidéo, réduction du nombre d'images transmises par seconde tout d'abord, par rapport à la scène lumineuse continue captée par l'objectif de la caméra; réduction de la définition verticale de chaque image aussi, inhérente à la structure en lignes; limitation de la résolution horizontale également, le nombre de points par ligne étant directement lié à la bande passante ; diminution de la définition de l'image encore, due cette fois à la structure en triplets de luminophores de l'écran du récepteur; réduction de la quantité

Chapitre I Notions De Base Sur L'image Et La Compression Numériques

d'informations de chrominance transmises enfin, compte tenu de l'incapacité de l'oeil à discerner des différences de couleurs dans les détails finis.

Apparue au début des années 1980 la numérisation des images vidéo conformément à la norme 4:2:2 engendre des volumes de données gigantesques, donc des débits extrêmement élevés au regard de la capacité relativement limitée des supports de stockage et surtout, de transmission existants. Pour donner un ordre d'idée, une image 800x600 codé sur 3 couleurs (chaque couleur est quantifiée sur 8 bits) occupe un espace de près de 1,38 Méga-octets (en ne considérant que la partie active), ce qui porte la seconde à 41,2 Méga-octets. Un CD-Rom de 650 Mo ne peut pas contenir plus de 16 secondes d'une vidéo non compressée, et un DVD de 4,7 Go même pas 2 minutes. La réduction, du débit du signal vidéo est un passage obligé pour autoriser le stockage d'une grande quantité d'images sur une bande magnétique ou un support informatique, et pour permettre la diffusion de ces images numériques dans un canal de fréquence conventionnel.

Le signal vidéo numérique renferme un volume de données beaucoup trop important pour être transmis ou enregistré par des machines économiquement abordables. La compression a pour but de trouver la manière la plus rationnelle de coder les images (en termes d'encombrement), tout en préservant au maximum leur contenu.

1.3.1 Principe de la compression de la vidéo

L'art de la compression numérique en vidéo est de supprimer certaines informations de l'image et d'en simplifier d'autres, tout en faisant en sorte que les modifications apportées échappent le plus possible à la perception humaine. Cela est d'autant plus difficile que le débit final que l'on cherche à obtenir est faible. Dans le cas d'une image fixe, les techniques de compression s'appuient sur une analyse du contenu de l'image et tirent profit de son organisation interne afin d'en éliminer les données redondantes. Par exemple, une image comporte forcément des plages uniformes plus ou moins grandes, composées de pixels identiques que l'on peut coder de manière compacte. Dans le cas d'une séquence vidéo animée, la compression peut exploiter le fait qu'il existe très souvent une grande similitude entre plusieurs images successives. La plupart du temps, la vitesse des mouvements est largement inférieure à la fréquence de rafraîchissement des images. Là aussi, une économie de données peut être réalisée. Par ailleurs, il faut savoir que le système visuel humain n'exploite pas la totalité des informations présentes sur une image. Il est moins sensible aux fins détails de l'image (qu'ils soient fixes ou en mouvement) qu'aux plages uniformes. De plus, il possède un pouvoir de perception bien plus faible dans les détails de couleurs que dans les détails de luminosité. Par conséquent, il existe sur l'image vidéo une grande quantité d'informations auxquelles notre œil n'accorde que peu d'importance et pour lesquelles il peut se contenter d'une représentation approximative.

1.3.2 Types de Compressions vidéo:

La compression vidéo fait appel à une variété d'algorithmes de codage qui exploitent les différents types de redondance de l'image. Le choix et l'allocation de ces algorithmes se fait en fonction des applications visées et des débits souhaités.

On distingue deux grandes catégories d'algorithmes de compression:

- Les algorithmes dits « **sans pertes** » (lossless en anglais) effectuent un traitement totalement transparent, permettant de retrouver intégralement les données d'origine à l'issue de la décompression. Malheureusement, ils ne conduisent à des taux de compression très faibles, en tout cas insuffisants pour la plupart des applications vidéo.
- Les algorithmes dits « **avec pertes** » (lossy en anglais) aboutissent à des taux de compression nettement supérieurs, mais imposent de négliger certaines informations de l'image, en tenant compte de sa nature et de notre perception visuelle. Si elle se fait dans des proportions limitées, l'élimination de ces informations peut passer inaperçue pour un observateur moyen; on parle alors de compression virtuellement transparente. Si, en revanche, la réduction de débit doit être réalisée dans des facteurs élevés.

1.3.3 Les redondances de l'image vidéo :

➤ **La redondance spatiale :**

Toute plage uniforme sur une image renferme des pixels identiques. Il est donc inutile de coder séparément chacun de ces pixels puisqu'un seul peut les caractériser tous. Il suffit de transmettre deux données, l'une représentant la valeur du pixel, l'autre étant le facteur de répétition.

➤ **La redondance temporelle :**

Dans une séquence vidéo, il existe une très forte corrélation entre les images successives. Les techniques d'estimation de mouvement permettent de coder une image par rapport à sa voisine, en prévoyant ses changements et ne transmettant que les informations relatives aux éléments qui se sont déplacés. L'élimination des redondances temporelles : peut conduire à des taux de compression très élevés. En revanche; le codage s'applique dans ce cas non plus à des images isolées, mais à des groupes d'images rendues indissociables, parce que décrites les unes en fonction des autres. Il est donc bien adapté à la diffusion d'un flux continu d'informations, mais se prête difficilement au montage, surtout si les groupes d'images sont longs.

➤ **La redondance subjective :**

L'exploitation de la redondance subjective fait appel à la notion de codage perceptuel, tirant parti des faiblesses de la vision humaine. Elle consiste à coder avec un nombre de bits limité les éléments de l'image jugés les moins significatifs pour notre œil. Cette pondération psycho-visuelle est obtenue lors de la phase de quantification non linéaire, qui introduit des pertes de résolution.

➤ **La redondance statistique :**

Il s'agit d'une notion purement mathématique: si certains codes reviennent plus fréquemment que d'autres, autant leur réserver les mots les plus courts. Cette opération, appelée codage entropique, n'entraîne aucune perte. La compression en vidéo met en œuvre une variété d'outils de codage s'appuyant sur les trois principaux suivants.

- Il est inutile de répéter un à un les pixels qui sont identiques sur une image.
- Si une image est très semblable à sa voisine, il suffit de ne transmettre que leurs différences.
- Certaines informations peu ou pas pertinentes pour notre système visuel peuvent être codées plus grossièrement, voire supprimées.

Pour exploiter les différents types de redondances observées en vidéo, les méthodes employées sont :

- redondance spatiale : transformée en Cosinus Discrète (DCT).
- redondance temporelle : estimation de mouvement.
- redondance subjective : quantification non linéaire.
- redondance statistique : codage entropique.

1.3.4 Quelques ordres de grandeurs :

Dans le cas d'une image fixe, on peut typiquement supprimer jusqu'à 70% des informations sans toucher à sa qualité. On dit alors qu'un taux de compression de 3:1 est totalement transparent. Si l'on tolère quelques pertes, généralement peu décelables par un œil non averti, le taux de compression peut atteindre 10:1. Dans le cas d'une séquence vidéo, on peut obtenir des facteurs de compression nettement supérieurs en tenant compte de la forte parenté entre images contiguës. Par exemple, pour les applications de diffusion, il est courant de recourir à des taux de compression compris entre 15:1 et 40:1. Le DVD compresse le signal vidéo dans un facteur d'environ 25.

➤ Les normes de compression vidéo

En un peu plus de dix ans, quatre standards de compression numérique des images ont été définis par des groupes de travail spécialisés, formés sous les auspices de l'ISO (International Standardization Organization): JPEG (Joint Photo-graphics Experts Group), MPEG-1 (Motion Pictures Experts Group), MPEG-2 et MPEG-4.

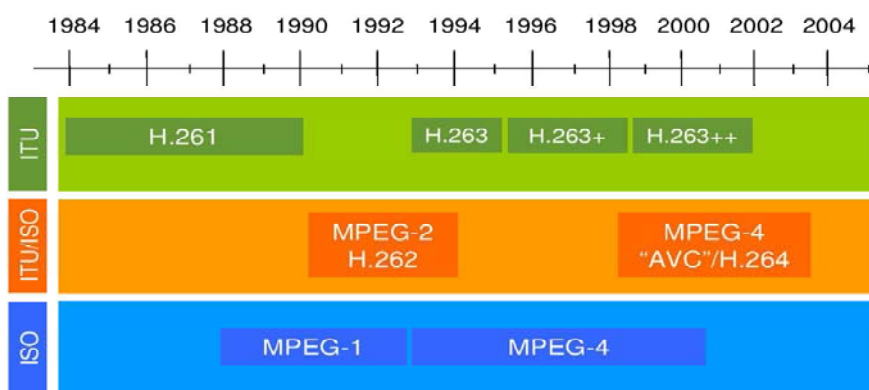


Figure 1 : Evolution des normes de compression vidéo.

➔ JPEG et M-JPEG :

Apparue en 1989, JPEG est une norme de compression des images fixes, conçue à l'origine pour le monde de l'impression et de la photocomposition. JPEG accepte n'importe quelle définition d'image et exploite les uniformités présentes à l'intérieur de chacune d'elle; le codage est dit « intra-image ». Or, comme les circuits intégrés de codage JPEG étaient disponibles bien avant tout autre circuit de compression, les fabricants d'équipements vidéo se sont rapidement intéressés à ce standard, même s'il ne leur était pas destiné au départ. Partant du principe qu'une séquence vidéo n'est qu'une succession rapide d'images fixes, ils ont développé des systèmes JPEG capables de compresser/décompresser en temps réel 25 ou 30 images par seconde. Ces solutions, baptisées M-JPEG (Motion-JPEG), se sont vite répandues à partir du début des années 90, notamment dans les stations de montage non-linéaire et dans le domaine de l'enregistrement haut de gamme. Cependant, les modifications ayant permis de passer du JPEG au M-JPEG, ainsi que la synchronisation du son, n'ont jamais fait l'objet d'une normalisation. Les fabricants ont donc développé sans concertation des solutions propriétaires, si bien que les fichiers générés par des équipements de marques différentes sont très souvent incompatibles entre eux.

L'échange des programmes M-JPEG n'est possible qu'à travers un cycle de compression/décompression du signal. Depuis, est apparue le format d'enregistrement DV, qui utilise une compression de type M-JPEG, mais dont l'algorithme a été totalement normalisé à l'échelle mondiale. Initialement développé pour le grand

Chapitre I Notions De Base Sur L'image Et La Compression Numériques

public avec un débit de 25.Mbits/s, le DV a progressivement évolué vers des débits supérieurs (50 et 100 Mbits/s), donnant naissance à une famille de systèmes de compression adaptés à tous les besoins de la chaîne de production et postproduction broadcast, en définition standard comme en haute définition: DV, DVCAM, DVCPRO25, DVCPRO50 et DVCPRO-HD.

→ MPEG-1:

Apparue en 1992, MPEG-1 est une norme de compression des images animées à faible résolution, destinée aux applications multimédia. Fruit de quatre années de travaux au sein du comité MPEG, le MPEG-1 a été conçu pour coder un programme audio/vidéo dans une qualité tout juste comparable à du VHS. Son débit maximal a en effet volontairement été bridé à 1,5 Mbits/s (dont 1,15 Mbits/s pour la vidéo), afin d'être compatible avec le stockage sur un Compact Disc (CD-Rom, CD-Vidéo), qui peut ainsi contenir 74 mn de programme. Pour parvenir à un débit aussi bas, MPEG-1 travaille sur une résolution d'image réduite à 1/4 seulement de celle d'une image standard (CIF 360 x 288), et ne prend en compte qu'une trame sur deux (l'autre est simplement répétée). Il effectue ensuite un codage intra-image de type JPEG à l'intérieur de chaque image, qu'il complète par un formidable codage inter-image. C'est ce dernier qui, en exploitant les redondances temporelles entre plusieurs images successives (rendues de ce fait indissociables) permet d'obtenir des taux de compression aussi élevés. L'une des applications phare du MPEG-1 est le CD-Vidéo, qui n'a pas véritablement percé en Europe, mais qui a rencontré un grand succès dans d'autres pays comme la Chine où plusieurs millions de lecteurs ont été vendus. MPEG-1 est aujourd'hui notamment utilisé comme format de consultation dans plusieurs applications broadcast, mais il tend à être remplacé par MPEG-4.

→ MPEG-2 :

Normalisé en 1994, MPEG-2 est devenu le standard de compression de référence pour tous les secteurs de diffusion/distribution audiovisuelles, et a permis à l'industrie grand public de migrer avec succès de la technologie analogique à la technologie Numérique. Il n'est pas exagéré de dire que toute image affichée sur un écran de télévision aujourd'hui a été au moins une fois codée en MPEG2 au cours de son cheminement. MPEG-2 reprend la philosophie et l'ensemble des techniques de base de MPEG-1 (compressions spatiale de type JPEG et temporelle), avec des outils optimisés pour le faire sérieusement monter en puissance en termes de débit, donc de qualité d'image. Tout d'abord, la résolution de l'image d'entrée s'étend de la définition standard (720 x 576 à la haute définition (1920 x 1080), avec un balayage entrelacé ou progressif. Le débit vidéo s'échelonne typiquement de 4 à 50.Mbits/s en SDTV, et peut atteindre 300 Mbits/s en HDTV. Du coup, MPEG3, qui devait se consacrer à part entière à la haute définition, a été totalement absorbé par MPEG-2, qui possède tous les outils nécessaires. MPEG-2 a d'abord été étudié pour la diffusion numérique; il est aujourd'hui mondialement utilisé par tous les opérateurs de télévision numérique et

constitue également le standard des DVD- Vidéo. Par la suite, MPEG-2 a été décliné en une version « studio », baptisée MPEG-2 422P@ML (ou MPEG-2 422), répondant aux exigences plus sévères de la production/postproduction broadcast et permettant de travailler en mode intra-image, comme en M-JPEG, pour permettre des manipulations et traitements complexes et précis des images.

→ MPEG-4 :

Normalisée en 2000, MPEG-4 se distingue fondamentalement de MPEG-1 et MPEG-2 par son approche orientée objet et l'intégration, dans une norme unique, de toutes les composantes liées à l'interactivité. Ce standard est capable de coder individuellement les différents éléments média d'une image (les « objets » audio, Vidéo, graphiques...) en les faisant bénéficier du traitement le mieux adapté à leur nature. Ces objets sont alors transmis séparément sur tout type de réseau, afin de permettre à l'utilisateur final d'interagir sur la composition de la scène audiovisuelle qu'il reçoit. En particulier, le traitement de la vidéo, qui a jusqu'à présent concentré l'essentiel des développements de la norme, est le plus performant de tous les standards actuels. Il reprend les grands principes déployés par MPEG-2; mais avec des outils plus sophistiqués et des techniques inédites, offrant un codage plus raffiné et aboutissant à une efficacité de compression très sensiblement supérieure. MPEG-4 est un standard extrêmement riche et complexe, pouvant fournir des débits allant de quelques Kbits/s à plusieurs centaines de Mbits/s. Son champ d'application est, on l'imagine, très vaste, allant du streaming sur Internet à l'enregistrement de la vidéo à haute définition, en passant par le multimédia mobile.

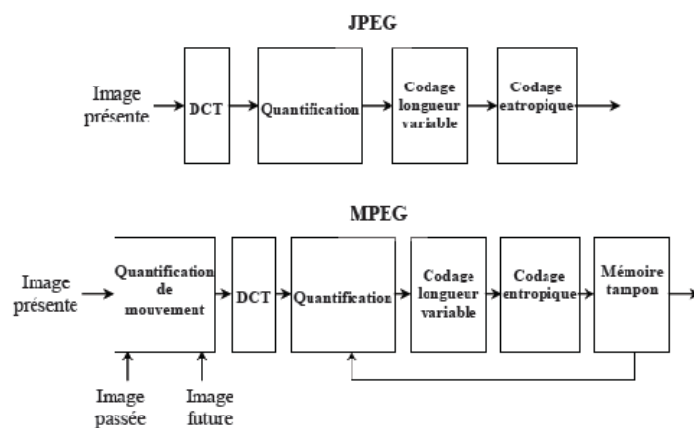


Figure I.2 : principe des compressions JPEG et MPEG.

La compression JPEG (et sa déclinaison vidéo M-JPEG) traite indépendamment chaque image en éliminant ses redondances spatiales : le codage est dit « intra-image ». Les compressions MPEG-1 et MPEG-2 traitent les images par groupes, en exploitant leurs redondances temporelles, c'est-à-dire les données communes à plusieurs images successives : le codage est dit « inter-image ». MPEG-2 peut cependant travailler uniquement en mode intra-image pour les applications de

studio. MPEG-4 reprend les principes de MPEG-1 et 2, mais avec des outils plus performants adaptés à tous les débits, et avec une approche orientée objet.

➤ **MPEG-1: La compression vidéo pour le multimédia:**

MPEG-1 est un standard de compression de séquences vidéo animées, associées à un son stéréo synchrone. MPEG-1 fournit un débit d'environ 1,5 Mbits/s (dont 1,15 Mbits/s, pour l'image), totalement calibré pour le CD-Rom. Avec une capacité de 650 Mo, celui-ci offre ainsi une capacité de 74 minutes de programme codé en MPEG-I. Pour parvenir à un taux de compression aussi élevé, MPEG-I reprend les techniques de JPEG pour ce qui est de l'élimination des redondances internes à chaque image, mais fait également appel à d'autres procédés, dont certains très radicaux. Tout d'abord, il travaille à partir d'une image source au format SIP (Source Inter-mediate Format), caractérisée par une réduction de moitié des résolutions spatiale et temporelle. La taille de l'image est d'1/4 d'écran et l'affichage s'effectue à 25 trames par seconde en balayage progressif. D'autre part, MPEG-I cherche à tirer pleinement parti des fortes similitudes entre les images. Il se contente de coder intégralement une image de temps en temps et d'indiquer, le reste du temps, uniquement quels éléments ont bougé.

Mis à part le choix du format d'entrée, toutes les techniques de compression utilisées par MPEG-1 ont été reprises par MPEG-2, avec cependant des paramètres différents garantissant une meilleure qualité.

En plus de l'analyse interne des images mise en œuvre dans JPEG, MPEG-1 travaille sur une image réduite à 25% de sa taille d'origine, ne traite qu'une trame sur deux, et élimine les redondances temporelles entre plusieurs images successives.

➔ **Les « GOP » en MPEG :**

Après avoir réduit les redondances spatiales à l'intérieur de chaque image au moyen de l'algorithme JPEG, la compression MPEG se poursuit en s'appliquant à des groupes d'images appelés GOP (Group Of Pictures): Les GOP se composent d'une combinaison de trois types d'images:

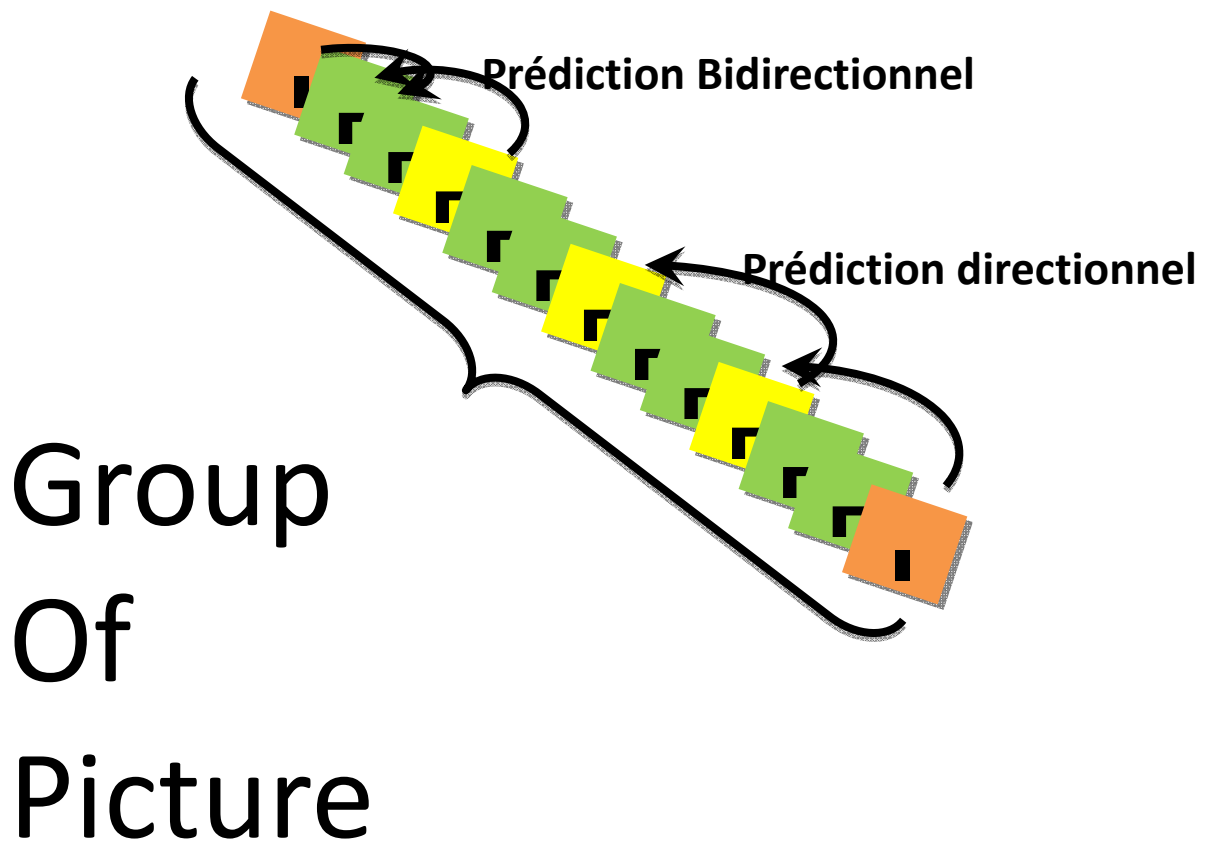
- **l'image I (Intra):** elle est codée en mode intra-image en JPEG. Elle est entièrement décrite par elle-même, sans aucune référence à d'autres, et contient tous les éléments nécessaires à sa reconstruction. Elle constitue de ce fait le point d'accès pour le décodage. La fréquence d'occurrence des images I dans une séquence MPEG conditionne la précision de l'accès aléatoire. Les images I bénéficient cependant d'un faible taux de compression et sont donc assez volumineuses.

- **l'image, P (Prédite):** elle est prédite à partir d'une image passée I ou P. Elle est codée uniquement à l'aide de vecteurs mouvement indiquant les déplacements de

ses éléments par rapport à l'image de référence. Une image P est typiquement trois fois moins volumineuse qu'une image I, mais peut transmettre des erreurs car elle sert également de référence. Il faut attendre l'arrivée d'une image I pour tout remettre à plat et relancer un nouveau processus de prédiction.

•**l'image B (Bidirectionnelle):** elle est construite, à l'aide de vecteur mouvement, par interpolation bidirectionnelle entre les images passées ou futures I ou P voisines. Elle offre le taux de compression le plus élevé, mais ne propage pas d'erreur, car elle n'est jamais utilisée en référence. Une image B est typiquement six fois moins volumineuse qu'une image I. C'est grâce aux images B que l'on peut faire chuter le débit d'un flux MPEG.

Un GOP commence toujours par une image I et se termine à la dernière image précédant la prochaine image I. Il peut se résumer à une seule image I, ou alors être composé d'une combinaison d'images I et P, I et B, ou encore I, B et P. La figure I.4 donne l'exemple d'un GOP contenant les trois types d'images I, P et B. On y relève deux paramètres fondamentaux: l'intervalle M séparant deux images Prédites et l'intervalle N entre deux images Intra (c'est la longueur du GOP). Les valeurs de M et N normalisées pour la diffusion sont : $M = 3$ et $N = 12$. Dans cette configuration, seule une image sur douze est transmise intégralement, soit deux par seconde; toutes les autres se réfèrent à leurs voisines.



Chapitre I Notions De Base Sur L'image Et La Compression Numériques

Figure 2 : La structure d'un GOP 13-2 (Nombre d'image : 13; la distance entre images I/P: 2).

Le nombre d'images Prédites séparant deux images Intra est ici assez élevé; ce qui implique de soigner le processus d'estimation de mouvement. La séquence type avec $M = 3$ et $N=12$ est la suivante :

I	B	B	P	B	B	P	B	B	P	B	B	I	B	B	P	...
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...

Notons que pour qu'une image B puisse être calculée, il faut que les images I et P dont elle dépend aient préalablement été reçues par le décodeur et conservées temporairement en mémoire. D'où la nécessité de modifier l'ordre de transmission des images par rapport a leur ordre naturel d'analyse. De ce fait, un retard de l'ordre de la durée du GOP est toujours introduit lors de la décompression. Cela donne, en reprenant notre exemple :

I	P	B	B	P	B	B	P	B	B	I	B	B	P	B	B	...
0	3	1	2	6	4	5	9	7	8	12	10	11	15	13	14	...

L'inconvénient majeur découlant d'un tel système de codage par longs groupes d'images est qu'il ne permet pas un accès direct à chaque image. On ne peut pas briser un GOP, car cela aurait pour conséquence de désolidariser une image P ou B de celle(s) servant à leur construction. Comme seul l'accès aux images I est autorisé, l'unité d'accès aléatoire vidéo en MPEG est le GOP. Il est par conséquent impossible ou disons qu'il serait extrêmement complexe d'effectuer du montage sur un programme ainsi codé, à moins de s'imposer des points d'entrée/sortie correspondant uniquement aux images I, soit, dans le cas présent, deux par seconde. En revanche, un GOP long présente l'avantage de permettre un taux de compression plus élevé qu'un GOP court pour une même qualité d'image. Autrement dit, à débit égal, un GOP long donne une meilleure qualité d'image qu'un GOP court. Dans les applications de diffusion et de distribution sur DVD, le GOP est respectivement de 12 et 15 images. En enregistrement broadcast cependant, il ne dépasse pas 2 images (format Betacam SX de Sony) et se réduit même, dans tous les autres cas, à une seule image.

→ **Le codeur MPEG :**

L'ordre des images est modifié par l'intermédiaire de mémoires de trame. Pour corriger les erreurs amenées par le calcul des vecteurs mouvements, une boucle assez complexe est introduite: les opérations de DCT et de quantification sont appliquées sur les images Intra avec la boucle de régulation de débit; puis les fonctions inverses sont réalisées (quantification inverse, DCT inverse, introduction des vecteurs mouvements) pour que l'image prédite par l'estimateur soit comparée avec la vraie nouvelle image, macro-bloc par macro-bloc. Une image de référence est alors produite, qui permettra au décodeur de rectifier les erreurs de prédiction éventuelles.

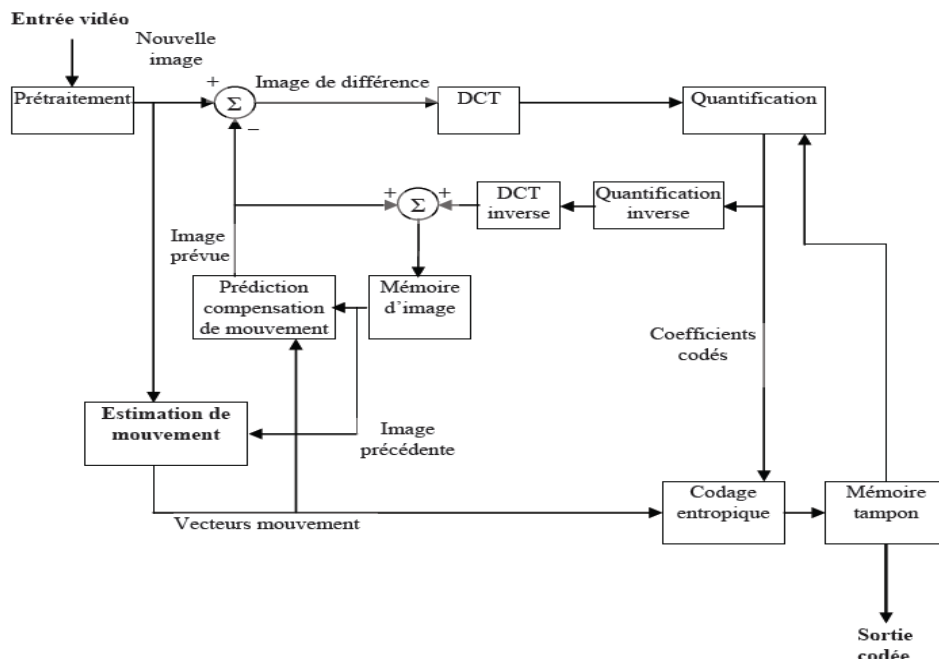


Figure I.4 : Synoptique du codeur MPEG.

→ Le décodeur MPEG :

Si le codeur est particulièrement complexe et onéreux, le décodeur est en revanche considérablement plus simple (il n'y a pas d'estimation de mouvement à faire) et relativement bon marché, ce qui est indispensable pour sa production à grande échelle sur le marché grand public. Après démultiplexage, les images Intra sont décodées - décodage à longueur variable, quantification inverse avec des modes programmés, DCT inverse. Les vecteurs de mouvement et les données de différence sont utilisés pour fabriquer les images Prédites à partir des images de référence. Les images Bidirectionnelles sont alors calculées, puis les images sont replacées dans leur ordre naturel.

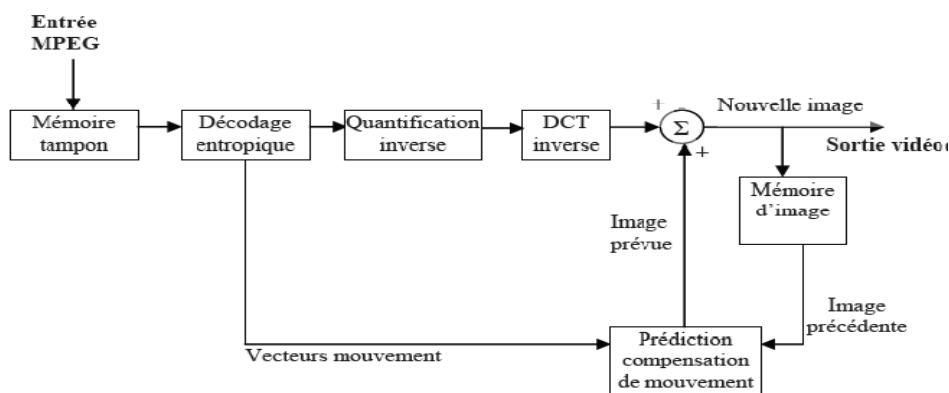


Figure I.4 : Synoptique du décodeur MPEG.

1.4. Mesure de la qualité vidéo

Dans n'importe quelle application vidéo (capture, stockage, transmission, visualisation) il est nécessaire de mesurer la qualité du signal. Cette nécessité est particulièrement importante dans le contexte de la vidéo streaming sur IP, où la distorsion est introduite au signal reçu à cause des pertes dues au réseau. Il y a deux approches principales pour mesurer la qualité d'une information visuelle :

L'approche subjective et l'approche objective. En bref l'approche subjective se base sur un groupe d'observateurs humains qui évaluent la qualité d'une image ou d'une séquence vidéo selon une échelle donnée. Par contre, l'approche objective utilise des techniques (généralement algorithmes de traitement de signal) pour produire une mesure automatique, quantitative, et répétable de la qualité visuelle.

1.4.1 Evaluation subjective :

Étant donné que la qualité vidéo est estimée au final par les utilisateurs du service, une mesure subjective, c'est-à-dire basée sur l'avis d'un échantillon de population, semble plus naturelle.

Une échelle de 1 à 5 est communément utilisée pour quantifier la distorsion perçue par les utilisateurs, comme le présente le tableau suivant.

Echelle	Qualité	Distorsion
5	Excellent	Imperceptible
4	Good	Perceptible sans être gênante
3	Fair	Légèrement gênante
2	Poor	Gênante
1	Bad	Très gênante

Tableau I.1: Echelle d'évaluation subjective.

Cette échelle est appelée **MOS** pour Mean Opinion Score.

Les observateurs constituent de préférence un groupe d'au minimum 15 personnes n'ayant pas de lien avec l'étude de la qualité vidéo. Ils doivent être informés des types de distorsions recherchées.

1.4.2 Evaluation objective :

Il y a plusieurs métriques objectives, le plus couramment utilisé est le PSNR (Peak Signal to Noise Ratio), qui est détaillé dans [ITU3, 15], [ANSI18, 19] et [MPEG9]. Une image peut être considérée comme un signal (image en niveaux de gris) ou un ensemble de signaux bidirectionnels (image couleur) $F(x, y)$. Si on considère

Chapitre I Notions De Base Sur L'image Et La Compression Numériques

une image $Y_S(x, y)$ de dimensions $N \times M$ et sa reconstituée après compression et décompression $Y_D(x, y)$, On définit alors le PSNR par :

$$PSNR = 20 \log \frac{V}{\sqrt{\frac{1}{NM} \sum_{i=0}^N \sum_{j=0}^M [Y_S(n, i, j) - Y_D(n, i, j)]^2}}$$

Avec $V = 2^k - 1$; k=nombre de bit par Pixel.

Le rapport $\sqrt{\frac{1}{NM} \sum_{i=0}^N \sum_{j=0}^M [Y_S(n, i, j) - Y_D(n, i, j)]^2}$ sert à calculé RMSE [RootMean Square Error] l'erreur quadratique moyenne.

Le PSNR est largement utilisés car sa complexité algorithmique est très faible. Néanmoins, il reflète en général assez mal la qualité réelle de la vidéo ou de l'image.

➤ **Conversion PSNR MOS :**

Vu que le PSNR n'est qu'une évaluation objective, il ne représente pas exactement le niveau de la qualité perçue par l'utilisateur final. Pour cela on effectue généralement une conversion PSNR-MOS. Le tableau 6.2 donne une conversion possible.

PSNR [dB]	MOS
> 37	5
31-37	4
25-31	3
20-25	2
< 20	1

Tableau 2: Conversion PSNR MOS.

1.4.3 Applications de la vidéo sur un Réseau IP :

➤ **Broadcast :**

Le broadcast ou La diffusion de vidéo sur IP est la transmission unidirectionnelle sur un réseau du contenu d'un fichier vidéo. Le terminal de lecture est un simple visionneur passif sans aucun contrôle sur la session. La diffusion de vidéo peut être Unicast ou Multicast à partir du serveur. Dans une configuration Unicast, le serveur répète la transmission pour chaque destinataire final. Dans une configuration Multicast, le même signal est envoyé sur le réseau, en une seule transmission, à plusieurs destinataires, ou simplement, à un groupe d'utilisateurs. Les entreprises font appel à cette technologie pour distribuer des formations, des présentations, des comptes-rendus de réunions et des discours. Elle est également utilisée par les universités, les centres de formation technique et de formation

continue, les diffuseurs de contenus, les fournisseurs de programmes télévisés sur le Web, etc.

➤ **Visioconférence :**

La visioconférence (VC) est une combinaison de transmissions audio et vidéo en mode full duplex. Elle permet à des personnes, éloignées géographiquement, de se voir et de s'entendre comme si elles étaient face à face. Les communications s'effectuent en temps réel et ne sont généralement pas enregistrées. Les services basés sur IP sont nettement plus performants. En effet, la conférence peut être amorcée à partir de n'importe quel PC du réseau équipé de l'équipement approprié. Les signaux transitent par les équipements et l'infrastructure réseau classique, éliminant de fait les lignes dédiées et les frais d'utilisation. Ces services peuvent être utilisés pour les communications d'entreprise, les applications de télémédecine et de télésanté, la formation, le e-learning, le télétravail et le service client. La visioconférence peut adopter une configuration point à point (entre deux utilisateurs) ou multipoint (plusieurs utilisateurs participant à une même session). Dans ce dernier cas, chaque utilisateur apparaît dans une fenêtre distincte. La visioconférence est à l'origine du nouveau concept de collaboration à distance.

➤ **Vidéo On Demand :**

De manière générale, la VOD permet à un utilisateur de visionner à la demande une vidéo en flux continu stockée sur un serveur. Cette technologie diffère de la diffusion de vidéo en ceci que l'utilisateur peut interrompre la lecture, la redémarrer, accélérer la vidéo ou revenir en arrière. Le service est interactif. La VOD possède une autre caractéristique: elle est généralement accompagnée de données qui indiquent l'utilisation effective de la vidéo, la facturation des services ou de la durée de la vidéo. Bien que la VOD puisse être visionnée en temps réel, elle est généralement utilisée pour visionner des fichiers vidéo stockés. Cette technologie est utilisée dans le cadre de sessions de e-learning, de formation, à des fins de marketing, de divertissement, de diffusion et dans tous les domaines où l'utilisateur préfère choisir à quel moment il souhaite visionner le contenu, sans dépendre des horaires fixés par le fournisseur.

1.5. Conclusion

La compression de la vidéo permet de limiter la taille de la vidéo en fonction de la taille des unités de stockage et d'adaptera le débit de la vidéo en fonction des technologies qui ne permettent pas de visualisation en non-compressé.

CHAPITRE II :

LES Réseaux Informatiques

1.1. Introduction :

Les réseaux de données sont utilisés essentiellement pour interconnecter des ordinateurs, leurs serveurs et les plates-formes qui les gèrent. Les applications utilisant un réseau de données sont généralement des applications informatiques assurant l'échange de données entre deux machines. Le déploiement des réseaux de données a explosé suite à l'introduction massive des ordinateurs dans les entreprises et les universités, vu le besoin croissant en matière d'échanges et de partages des informations.

Nous aborderons dans ce chapitre l'un des modèles de réseaux de données le plus utilisé au monde, en l'occurrence le TCP/IP, à travers les points suivants :

- Les couches de TCP/IP.
- Le protocole TCP,
- Le protocole IP,
- L'adressage IP,
- Les protocoles de routage,
- La sécurité dans les réseaux IP,
- L'évolution des réseaux IP, IPv6.

1.2. Le modèle OSI**➔ *Présentation du modèle OSI***

OSI signifie Open Systems Interconnexion, ce qui se traduit par Interconnexion de systèmes ouverts. Ce modèle a été mis en place par l'ISO afin de mettre en place un standard de communications entre les ordinateurs d'un réseau, c'est-à-dire les règles qui gèrent les communications entre des ordinateurs. En effet, aux origines des réseaux chaque constructeur avait un système propre (on parle de système propriétaire). Ainsi de nombreux réseaux incompatibles coexistaient. C'est la raison pour laquelle l'établissement d'une norme a été nécessaire.

Le rôle du modèle OSI consiste à standardiser la communication entre les machines afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles (pour peu qu'ils respectent scrupuleusement le modèle OSI).

→ **Le modèle en couche OSI**

Le modèle OSI est un modèle qui comporte 7 couches qui se sont les suivantes:



Figure II.1 : Le modèle OSI

- **La couche physique** définit la façon dont les données sont physiquement converties en signaux numériques sur le média de communication (impulsions électriques...etc.).
- **La couche liaison données** définit l'interface avec la carte réseau et le partage du média de transmission.
- **La couche réseau** permet de gérer l'adressage et le routage des données, c'est-à-dire leur acheminement via le réseau.
- **La couche transport** est chargée du transport des données, de leur découpage en paquets et de la gestion des éventuelles erreurs de transmission.
- **La couche session** définit l'ouverture et la destruction des sessions de communication entre les machines du réseau.
- **La couche présentation** définit le format des données manipulées par le niveau applicatif (leur représentation, éventuellement leur compression et leur chiffrement) indépendamment du système.
- **La couche application** assure l'interface avec les applications. Il s'agit donc du niveau le plus proche des utilisateurs, géré directement par les logiciels.

I.3. Le modèle TCP/IP :

Les réseaux de données sont caractérisés par une certaine neutralité par rapport aux applications supportées. C'est-à-dire que l'architecture du réseau n'est pas conditionnée par une application spécifique. Ceci offre l'avantage d'avoir un réseau multiservice et surtout évolutif, pouvant servir ultérieurement pour des applications futures TCP/IP, du nom de ses deux protocoles principaux (TCP, Transmission Control Protocol et IP, Internet Protocol), est un ensemble de protocoles permettant de

résoudre les problèmes d'interconnexion en milieu hétérogène. A cet effet, TCP/IP décrit un réseau logique (réseau IP) au-dessus des réseaux physiques réels auxquels sont effectivement connectés les ordinateurs.

TCP/IP utilise le principe de l'encapsulation qui consiste à transporter les données d'une couche dans une unité de données de la couche inférieure. Un en-tête contient les informations nécessaires à l'entité homologue pour extraire et traiter les données. Dans le modèle TCP/IP, les données de l'application constituent des messages, ceux-ci sont transportés dans des segments qui seront émis sur le réseau sous forme de datagrammes. L'unité de transport élémentaire est la trame qui constitue au niveau physique un train de bits.

➔ **Le model en couche :**

Le modèle TCP/IP contient 4 couches :

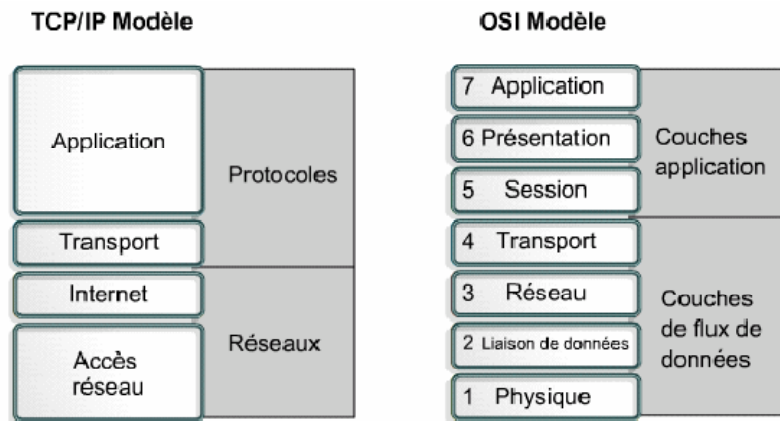


Figure II.1 Comparaison entre modèle OSI et TCP/IP.

- **La couche accès réseau :** elle regroupe deux sous-couches :

- ❖ La sous-couche physique : la couche physique décrit les caractéristiques physiques de la communication, comme les conventions à propos de la nature du médium utilisé pour les communications (les câbles, les liens par fibre optique ou par radio), et tous les détails associés comme les connecteurs, les types de codage ou de modulation, le niveau des signaux, les longueurs d'ondes, la synchronisation et les distances maximales.

- ❖ La sous-couche liaison de données : elle spécifie comment les paquets sont transportés sur la couche physique, et en particulier le tramage (i.e. les séquences de bits particulières qui marquent le début et la fin des paquets). Les en-têtes des trames Ethernet, par exemple, contiennent des champs qui indiquent à quelle(s) machine(s) du réseau un paquet est destiné. Exemples de protocoles de la couche de liaison de données : Ethernet, Wireless Ethernet, SLIP, Token Ring et ATM. PPP est

un peu plus protocole de liaison de données. Cette couche est parfois subdivisée en LLC et MAC.

- **La couche Internet** : elle permet l'acheminement de données depuis un réseau source vers un réseau destinataire. Ceci implique généralement le routage des paquets à travers le réseau Internet. Dans la suite de protocoles Internet, IP assure l'acheminement des paquets depuis une source vers une destination, et supporte aussi d'autres protocoles

- **La couche transport** : sur laquelle s'appuient directement les applications, fournit deux types de service : un service en mode connecté (TCP), et un service de transport allégé UDP (User Datagram Protocol) qui n'offre qu'un service de type best effort (datagramme).

- **La couche application** : c'est dans la couche application que se situent la plupart des programmes réseau.

1.4. Les principaux protocoles et applications de l'environnement TCP/IP :

- HTTP, HyperText Transport Protocol, assure le transfert de fichiers hypertextes entre un serveur Web et un client Web.

- FTP, File Transfert Protocol, est un système de manipulation de fichiers à distance (transfert, suppression, création...).

- TELNET, TELtypewriterNETworkprotocol ou TERminalNETworkprotocol, système de terminal virtuel, permet l'ouverture de sessions avec des applications distantes.

- SMTP, Simple Mail Transfert Protocol, offre un service de courrier électronique

- TFTP, Trivial FTP, est une version allégée du protocole FTP.

- DNS, Domain Name Server, est un système de bases de données réparties assurant la correspondance d'un nom symbolique et d'une adresse Internet (adresse IP)

- RIP, Routing Information Protocol, est le premier protocole de routage (vecteur distance) utilisé dans Internet

- SNMP, Simple Network Management Protocol, est devenu le standard des protocoles d'administration de réseau
- ICMP, Internet Control and error Message Protocol, assure un dialogue IP/IP et permet notamment : la signalisation de la congestion, la synchronisation des horloges et l'estimation des temps de transit. Il est utilisé par l'utilitaire PING qui permet de tester la présence d'une station sur le réseau
- ARP, AdressResolution Protocol, est utilisé pour associer une adresse logique IP à une adresse physique MAC (Medium Access Control, adresse de l'interface dans les réseaux locaux)
- RARP, Reserve AdressResolution Protocol, permet l'attribution d'une adresse IP à une station.
- OSPF, Open ShortestPath First, est un protocole de routage du type état des liens, il succède à RIP.
- SLIP, Serial Line Interface Protocol, protocole d'encapsulation des datagrammes IP, il assure la délimitation des trames.
- PPP, Point to Point Protocol, protocole d'encapsulation des datagrammes IP, il assure la délimitation des trames, identifie le protocole transporté et la détection d'erreurs.

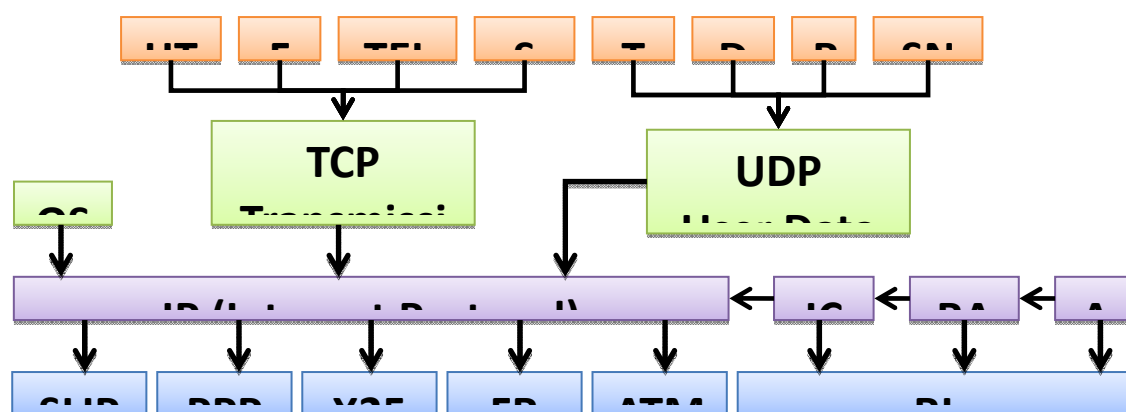


Figure II.2 Protocoles et applications du model TCP/IP.

1.5. Le protocole TCP :

➔ Les caractéristiques du protocole TCP

TCP est un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle. Lorsque les données sont fournies au

protocole IP, celui-ci les encapsule dans des datagrammes IP, en fixant le champ protocole à 6 (Pour savoir que le protocole en amont est TCP...). TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission.

Les caractéristiques principales du protocole TCP sont les suivantes:

- TCP permet de remettre en ordre les datagrammes en provenance du protocole IP.
- TCP permet de vérifier le flot de données afin d'éviter une saturation du réseau.
- TCP permet de formater les données en segments de longueur variable afin de les "remettre" au protocole IP.
- TCP permet de multiplexer les données, c'est-à-dire de faire circuler simultanément des informations provenant de sources (applications par exemple) distinctes sur une même ligne.
- TCP permet enfin l'initialisation et la fin d'une communication de manière courtoise.

→ Le format des données sous TCP

Un segment TCP est constitué comme suit:

0		15						16																32																							
Port source								PORT Destinataire																																							
Numéro de Séquence																																															
Accusé se réception																																															
Data offset	réservé	U	A	P	R	S	F	Fenêtre																																							
Checksum																Pointeurs donnés urgents																															
Option																Bourrage																															
Données																																															

Figure 1.8 : Format de segment TCP

Signification des différents champs:

- **Port Source** (16 bits): Port relatif à l'application en cours sur la machine source
- **Port Destination** (16 bits): Port relatif à l'application en cours sur la machine de destination
- **Numéro d'ordre** (32 bits): Lorsque le drapeau SYN est à 0, le numéro d'ordre est celui du premier mot du segment en cours. Lorsque SYN est à 1, le numéro

d'ordre est égal au numéro d'ordre initial utilisé pour synchroniser les numéros de séquence (ISN)

- **Numéro d'accusé de réception** (32 bits): Le numéro d'accusé de réception également appelé numéro d'acquittement correspond au numéro (d'ordre) du prochain segment attendu, et non le numéro du dernier segment reçu.
- **Décalage des données** (4 bits): il permet de repérer le début des données dans le paquet. Le décalage est ici essentiel car le champ d'options est de taille variable.
- **Réservé** (6 bits): Champ inutilisé actuellement mais prévu pour l'avenir
- **Drapeaux (flags)** (6x1 bit): Les drapeaux représentent des informations supplémentaires:
 - **URG**: si ce drapeau est à 1 le paquet doit être traité de façon urgente.
 - **ACK**: si ce drapeau est à 1 le paquet est un accusé de réception.
 - **PSH (PUSH)**: si ce drapeau est à 1, le paquet fonctionne suivant la méthode PUSH.
 - **RST**: si ce drapeau est à 1, la connexion est réinitialisée.
 - **SYN**: Le Flag TCP SYN indique une demande d'établissement de connexion.
 - **FIN**: si ce drapeau est à 1 la connexion s'interrompt.
- **Fenêtre** (16 bits): Champ permettant de connaître le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception
- **Somme de contrôle** (Checksum ou CRC): La somme de contrôle est réalisée en faisant la somme des champs de données de l'en-tête, afin de pouvoir vérifier l'intégrité de l'en-tête
- **Pointeur d'urgence** (16 bits): Indique le numéro d'ordre à partir duquel l'information devient urgente
- **Options** (Taille variable): Des options diverses
- **Remplissage**: On remplit l'espace restant après les options avec des zéros pour avoir une longueur multiple de 32 bits.

1.6. Le protocole UDP**→ Les caractéristiques du protocole UDP**

Le protocole UDP (User Datagram Protocol) est un protocole non orienté connexion de la couche transport du modèle TCP/IP. Ce protocole est très simple étant donné qu'il ne fournit pas de contrôle d'erreurs (il n'est pas orienté connexion...).

L'en-tête du segment UDP est donc très simple:

Port UDP source	Port UDP destination
Longueur	Checksum
données	

Figure 1.13 : Format de segment UDP

Signification des différents champs

- **Port Source**: il s'agit du numéro de port correspondant à l'application émettrice du segment UDP. Ce champ représente une adresse de réponse pour le destinataire. Ainsi, ce champ est optionnel, cela signifie que si l'on ne précise pas le port source, les 16 bits de ce champ seront mis à zéro, auquel cas le destinataire ne pourra pas répondre (cela n'est pas forcément nécessaire, notamment pour des messages unidirectionnels).
- **Port Destination**: Ce champ contient le port correspondant à l'application de la machine destinataire à laquelle on s'adresse.
- **Longueur**: Ce champ précise la longueur totale du segment, en-tête comprise, or l'en-tête a une longueur de 4 x 16 bits (soient 8 x 8 bits) donc le champ longueur est nécessairement supérieur ou égal à 8 octets.
- **Somme de contrôle**: Il s'agit d'une somme de contrôle réalisée de telle façon à pouvoir contrôler l'intégrité du segment.

1.7. Le protocole IP**→ Le rôle du protocole IP**

Le protocole IP fait partie de la couche Internet de la suite de protocoles TCP/IP. C'est un des protocoles les plus importants d'Internet car il permet l'élaboration et le transport des datagrammes IP (les paquets de données), sans toutefois en assurer la « livraison ». En réalité, le protocole IP traite les datagrammes IP indépendamment les uns des autres en définissant leur représentation, leur routage et leur expédition.

Le protocole IP détermine le destinataire du message grâce à 3 champs:

- Le champ adresse IP : adresse de la machine
- Le champ masque de sous-réseau : un masque de sous-réseau permet au protocole IP de déterminer la partie de l'adresse IP qui concerne le réseau
- Le champ passerelle par défaut : Permet au protocole Internet de savoir à quelle machine remettre le datagramme si jamais la machine de destination n'est pas sur le réseau local

➔ **Les datagrammes**

Les données circulent sur Internet sous forme de datagrammes (on parle aussi de paquets). Les datagrammes sont des données encapsulées, c'est-à-dire des données auxquelles on a ajouté des en-têtes correspondant à des informations sur leur transport (telles que l'adresse IP de destination). Les données contenues dans les datagrammes sont analysées (et éventuellement modifiées) par les routeurs permettant leur transit.

Voici ce à quoi ressemble un datagramme:

0	3	4	7	8	15	16	31
version		Lg. en-tête			Type de service		Longueur totale
Identification						Flags	Fragment offset
Durée de vie				Protocole		Checksum header	
Adresse source							
Adresse destination							
Option+bourrage							
Données							

Figure 1.4 : Format de datagramme IP

Voici la signification des différents champs:

- **Versio**n (4 bits) : il s'agit de la version du protocole IP que l'on utilise (actuellement on utilise la version 4 IPv4) afin de vérifier la validité du datagramme. Elle est codée sur 4 bits.
- **Longueur d'en-tête**, ou IHL pour Internet Header Length (4 bits) : il s'agit du nombre de mots de 32 bits constituant l'en-tête.
- **Type de service** (8 bits) : il indique la façon selon laquelle le datagramme doit être traité.
- **Longueur totale** (16 bits): il indique la taille totale du datagramme en octets. La taille de ce champ étant de 2 octets, la taille totale du datagramme ne peut dépasser 65536 octets. Utilisé conjointement avec la taille de l'en-tête, ce champ permet de déterminer où sont situées les données.

- **Identification, drapeaux (flags) et déplacement de fragment** sont des champs qui permettent la fragmentation des datagrammes, ils sont expliqués plus bas.
- **Durée de vie** appelée aussi **TTL**, pour Time To Live (8 bits) : ce champ indique le nombre maximal de routeurs à travers lesquels le datagramme peut passer. Ainsi ce champ est décrémenté à chaque passage dans un routeur, lorsque celui-ci atteint la valeur critique de 0, le routeur détruit le datagramme. Cela évite l'encombrement du réseau par les datagrammes perdus.
- **Protocole** (8 bits) : ce champ, en notation décimale, permet de savoir de quel protocole est issu le datagramme
 - ICMP : 1
 - IGMP : 2
 - TCP : 6
 - UDP : 17
- **header checksum (16 bits)** : ce champ contient une valeur codée sur 16 bits qui permet de contrôler l'intégrité de l'en-tête afin de déterminer si celui-ci n'a pas été altéré pendant la transmission. La somme de contrôle est le complément à un de tous les mots de 16 bits de l'en-tête (champ somme de contrôle exclu). Celle-ci est en fait telle que lorsque l'on fait la somme des champs de l'en-tête (somme de contrôle incluse), on obtient un nombre avec tous les bits positionnés à 1
- **Adresse IP source** (32 bits) : Ce champ représente l'adresse IP de la machine émettrice, il permet au destinataire de répondre
- **Adresse IP destination** (32 bits) : adresse IP du destinataire du message.

1.8. Protocoles de transmission Temps réel :

La contrainte temps réel est une exigence plus importante qu'un transfert en ordre et fiable pour les applications multimédia. Par conséquent le protocole UDP est généralement préféré que TCP, particulièrement pour les applications sensibles au délai comme la vidéoconférence et la voix sur IP. Pour mettre en œuvre des caractéristiques telles que la détection des pertes, la signalisation des erreurs et la synchronisation des flux multimédia, un protocole plus sophistiqué est nécessaire.

→ Le protocole RTP :

Le groupe de l'IETF a développé en 1993 le protocole de transport en temps réel (RTP, RFC 1889) dont le but est de transmettre sur Internet des données qui ont des propriétés temps réel (audio, vidéo ...etc.). C'est un protocole de la couche application du modèle OSI et utilise les protocoles de transport TCP ou UDP, mais, généralement, il utilise UDP qui est mieux approprié à ce genre de transmission. Le

rôle principal de RTP consiste à mettre en œuvre des numéros de séquences de paquets IP et des mécanismes d'horodatages (timestamp) pour permettre de reconstituer les informations de voix ou vidéo. Plus généralement, RTP permet :

- D'identifier le type de l'information transportée;
- D'ajouter des indicateurs de temps (horodater) et des numéros de séquence à l'information transportée;
- De contrôler l'arrivée à destination des paquets.

L'en-tête RTP (Tableau 1.1) indique le type, la source, et les caractéristiques de contrôle du temps des données encodées. Ce standard inclut en effet un horodateur (Time-stamp) des paquets permettant au destinataire d'utiliser la numérotation des séquences pour reconstituer l'ordre original des paquets et de détecter les paquets perdus.

V	P	X	CC	M	PT	Numéro de séquence
Horodatage (Timestamp)						
Identificateur de source de synchronisation (SSRC)						
Identification des sources contributrices (CSRC)						
En-tête supplémentaire						
Données						

Figure II.7 : En-tête du paquet RTP

- Le champ **V** (2 bits) : Version RTP, actuellement V=2.
- Le champ **P** (1 bit) : Si P=1 le paquet contient des octets additionnels de bourrage (padding) pour finir le dernier paquet.
- Le champ Extension **X** (1 bit) : Si X=1, l'en-tête est suivie d'un paquet d'extension.
- Le champ **CC (CSRC Count)** (4 bits) : Il contient le nombre de CSRC qui suivent l'en-tête.
- Le champ **M** (1 bit) : c'est le bit de marquage, son interprétation dépend de l'application (il sert lorsque l'on doit repérer des endroits particuliers dans le flux de données).
- Le champ type de la charge utile du paquet **PT** (7 bits) : Ce champ identifie le type de données (audio, vidéo, image, texte...).
- Le champ numéro de séquence (16 bits) : Sa valeur initiale est aléatoire et il s'incrémente de 1 à chaque paquet envoyé, il peut servir à détecter des paquets perdus.
- Le champ horodatage (32 bits) : Ce champ reflète l'instant où le premier octet du paquet RTP a été échantillonné.
- Le champ **SSRC** (32 bits) : Ce champ identifie de manière unique la source de synchronisation. Sa valeur est choisie de manière aléatoire par l'application.
- Le champ **CSRC** (32 bits) : Ce champ identifie les sources participantes.

RTP supporte différents types de codage et de compression des données. Beaucoup de formats standardisés sont acceptés (GSM, G.723, G.729 pour l'audio et MPEG-2, MPEG-4, H261, H263...etc. pour la vidéo).

➔ **Le protocole RTCP :**

RTCP (Real Time Control Protocol, RFC 1889) est un protocole de contrôle utilisé conjointement avec RTP pour contrôler les flux de données et la gestion de la bande passante.

RTCP permet de contrôler le flux RTP, et de véhiculer périodiquement des informations de bout en bout pour renseigner sur la qualité de service de la session de chaque participant à la session. Des quantités telles que le délai, la gigue, les paquets reçus et perdus sont très importants pour évaluer la qualité de service de toute transmission et réception temps réelles.

C'est le protocole sous-jacent (UDP par exemple) qui permet grâce à des numéros de ports différents et consécutifs (port pair pour RTP et port impair immédiatement supérieure pour RTCP) le multiplexage des paquets de données RTP et des paquets de contrôle RTCP. Le protocole RTCP remplit trois fonctions :

- L'information sur la qualité de service : RTCP fournit, en rétroaction des informations sur la qualité de réception des données transmises dans les paquets RTP. Cette information est utilisée par la source émettrice pour adapter le type de codage au niveau des ressources disponibles.
- L'identification permanente : RTCP transporte un identificateur original de la source RTP c'est à dire la provenance du flux, appelé CNAME (Canonical name). Cet identificateur permet une identification permanente de chacun des flux multimédia entrants.
- Le calibrage de la fréquence d'émission : La réception des feed-back et la connaissance du nom permanent servent à ajuster la fréquence d'envoi des paquets à la bande passante mise à la disponibilité de l'utilisateur situé à l'autre extrémité.

Chaque paquet RTCP contient un rapport émetteur ou récepteur suivi d'une description de la source (source description).

Il existe cinq types de paquets de contrôle (Tableau 1.2). Chaque paquet commence par un en-tête fixe suivi d'éléments structurés qui peuvent être de longueur variable selon le type de paquet.

RR	Rapport Récepteur
SR	Rapport Émetteur
SDES	Description de Source
BYE	Fin de session
APP	Nouveau champ

Tableau 1.2 : Messages RTCP

- **RR** : contient le rapport de la qualité de la livraison des données des participants passifs (récepteur) incluant le nombre de paquets reçus, le nombre de paquets perdus, la gigue et l'horodatage qui permet le calcul du délai total de transmission entre les deux parties.
- **SR** : contient le rapport de la qualité de livraison des données des participants actifs (émetteurs). Il contient les champs du RR, et des informations sur l'émetteur, la synchronisation (pour synchroniser deux sources de données), un compteur cumulatif de paquets et le nombre d'octets envoyés.
- **SDES** : contient l'information concernant les émetteurs comme le nom canonique (CNAME), et le nom d'utilisateur (NAME), le numéro de téléphone (PHONE) et d'autres informations concernant les participants.
- **BYE** : indique la fin de la participation d'une des parties.
- **APP** : Dans le cas des applications spécifiques, les données peuvent être transmises dans des paquets spécifiques de type APP.

Le tableau 1.3 détaille l'en-tête des paquets RTCP :

V	P	RC	PC	Longueur
Rapport(s)				

Tableau 1.3 : En-tête des paquets RTCP.

Avec :

- Le champ **V** (2 bits) : Indique la version de RTP, actuellement V=2.
- Le champ **P** (1 bit) : Si P=1 le paquet contient des octets additionnels de bourrage (padding) pour finir le dernier paquet.
- Le champ **RC** (5 bits) : Il contient le nombre de rapports contenus dans le paquet (un paquet pour chaque source).
- Le champ **PT** (8 bits) : Il donne le type de rapport du paquet (PT=SR=200, PT=RR=201, PT=SDES=202, PT=BYE=203, PT=APP=204).
- Le champ Longueur (16 bits) : Il indique la longueur du paquet.

1.9. Conclusion

Dans ce chapitre, nous avons classé les différents protocoles pour les réseaux informatiques pour la transmission et on a distingué deux types de transmission UDP pour la transmission en temps réel et TCP pour assurer la transmission.

CHAPITRE III

QUALITE DE SERVICE DANS UN RESEAU IP

III.1. La Qualité de service dans les réseaux IP

Avec l'émergence des réseaux IP, les réseaux informatiques sont omniprésents. Chaque entreprise, université, administration, possède son propre réseau. Ces derniers sont de plus en plus connectés à Internet développant au fil des années de nouvelles applications. Internet est utilisé en grande partie pour la messagerie électronique et la visualisation de page Web. Aujourd'hui de nouvelles applications orientées multimédia, communication tel que la téléphonie sur IP, la vidéoconférence, ou encore la visualisation de vidéo via streaming s'ajoute au trafic dit classique circulant sur les réseaux. On parle aujourd'hui de Triple Play, c'est la convergence des applications voix, vidéo et données. Tous ces nouveaux services requièrent une bande passante importante. Ils utilisent le protocole IP, ne garantissant pas au trafic d'arriver à destination. La solution serait de pouvoir définir le comportement du réseau en fonction du trafic observé pour être en mesure d'utiliser ces nouveaux services. Afin de garantir un niveau de performance et de qualité en fonction de l'application utilisée, la qualité de service a été développée, abrégée par l'acronyme QoS (Quality of Service). Ce chapitre traite, l'importance de la QoS, les critères de la QoS, et les approches standardisées par l'IETF afin de garantir cette qualité.

III.1.1. Importance de la qualité de service :

A l'heure actuelle, la QoS est utilisée par les entreprises utilisant les applications de type Triple Play, soucieuses de rentabiliser leur installation réseau tout en ayant une qualité élevée. Malgré l'existence de plusieurs protocoles et l'investissement des constructeurs, la QoS reste inexploitée. Les vitesses de transmission et les performances des réseaux connaissant un développement accru, on pourrait se demander en quoi la mise en place de la QoS est importante. Pourquoi ne pas continuer à sur-dimensionner les réseaux comme le font les opérateurs, et ainsi diminuer les congestions dans les liens. Cette solution serait à terme trop coûteuse et ne permettrait pas une utilisation intelligente des équipements. La QoS permet d'utiliser l'infrastructure réseau préexistante sans surdimensionner le réseau et engendrer des dépenses supplémentaires. Afin de garantir une qualité de service, il est important de différencier plusieurs types de trafic et de ne pas offrir les mêmes performances et services pour chaque application. D'autre part, la Qualité de Service s'applique sur tout le réseau. L'objectif est d'avoir une qualité autant en périphérie des réseaux (entreprise) qu'en communication inter réseaux (Internet Service Provider ou opérateur). Pour que la QoS soit optimale, elle doit être mise en place de bout en bout. Les applications en Temps Réels comme la voix sur IP ou la vidéo requièrent des délais de transmission faible, de gigue et des exigences en termes de bande passante. Les données étant transportées dans des trames IP, si la trame 2 arrive avant la trame 1, l'image ou le son ne sera pas à l'arrivée identique à l'original.

La qualité de service permet :

- de s'adapter de façon économique aux besoins de l'entreprise en termes d'allocation de bande passante, et de garantir un niveau minimal de fonctionnement même en période de congestion.
- de mettre en place une politique de priorité différente de celle du protocole IP (Best Effort), afin d'offrir des services de type Triple Play de qualité.

III.1.2. Critères de QoS dans l'Internet :

En effet, la caractérisation de la qualité de service dans l'Internet est généralement exprimée par les critères suivants :

• **Le délai** : c'est le temps écoulé entre l'envoi d'un paquet par l'émetteur et sa réception par le destinataire, le délai tient compte du délai de propagation le long du chemin parcouru par les paquets et du délai de transmission induit par la mise en file d'attente des paquets dans les systèmes intermédiaires (les routeurs). La plupart des applications et surtout les applications temps réel sont très sensibles aux valeurs élevées de délais et exigent un délai limité pour qu'elles puissent fonctionner correctement.

• **La Gigue** : la gigue est la variation de délai de bout en bout. Certaines applications (audio et vidéo) subissent une distorsion du signal avec de grandes valeurs de gigue.

• **Le débit** : c'est en fait le taux de transfert maximum pouvant être maintenu entre deux points terminaux (un émetteur et un récepteur), ce facteur est influencé non seulement par les capacités physiques des liens, mais aussi par les autres flux partageant ces liens.

• **La fiabilité** : c'est le taux moyen d'erreur d'une liaison, actuellement les supports de transmission deviennent de plus en plus fiables et ce taux devient de plus en plus négligeable par rapport aux pertes dû aux situations de congestion des réseaux.

• **Le taux de perte** : les pertes sont généralement dues soit à la non fiabilité des liens physiques soient aux mécanismes de gestion des files d'attente dans les routeurs intermédiaires qui se trouvent souvent obligés d'éliminer des paquets pour faire face à des situations de congestion. Ce facteur est très important. En effet, certaines applications ne tolèrent pas des valeurs élevées de ce facteur.

Extrait de (ITU, 2001), le Tableau 2.1 illustre les principales applications multimédia ainsi que leurs paramètres de QoS.

Type de Média	Application	Communication	Taille des données	Délai	Gigue	Perte
Audio	Conversation	bidirectionnelle	4—64 Kbits/s	< 150 ms. préférée ms < 400. limite	< 1 ms	< 3% PLR
Audio	Streaming Audio	unidirectionnelle	16— 128 Kbits/s	< 10 s	< 1 ms	< 1% PLR
Vidéo	Vidéoconférence	bidirectionnelle	16— 384 Kbits/s	< 150 ms. Préférée < 400ms. limite	Synchro < 80 ms	< 1% PLR
Vidéo	Streaming Vidéo	unidirectionnelle	16— 384 Kbits/s	< 10s	N/A	< 1% PLR
<ul style="list-style-type: none"> - PLR: taux de paquets perdus ou PacketLost Ratio - BER taux d'erreurs bits ou Bit Error Rate la valeur exacte dépend beaucoup du codec utilisé Synchro: synchronisation entre audio et vidéo (lips-synchro) 						

Tableau III.1 : les principales applications multimédia et leurs paramètres de QoS

III.1. Les mécanismes de QoS sur IP :

Pour fournir de la QoS dans les réseaux IP plusieurs mécanismes ont été envisagés comme IntServ, et DiffServ ce dernier sera l'objet du chapitre suivant).

III.1.1. Integrated Services et signalisation RSVP :**→ Fonctionnement de IntServ :**

L'approche IntServ est basée sur l'idée d'allocation des ressources dans le réseau pour donner à chaque application les garanties dont elle a besoin, donc un contrôle d'admission doit avoir lieu. IntServ repose sur RSVP pour établir et supprimer les réservations de bande passante. On distingue trois classes de service :

- **Best Effort (BE) :** ce sont les applications élastiques, ou non temps réels, qui n'offrent pas de garantie de délai, par exemple : connexion à distance, transfert de fichiers, messagerie...etc.
- **ControlledLoad (CL) :** pour les services temps réels tolérant une variation du délai de transit. Le CL correspond à un service Best Effort sur un réseau non chargé, cela revient à donner à l'utilisateur des garanties sur le débit disponible.
- **Guaranteed Service (GS) :** offre à la fois un contrôle du délai de bout en bout pour les applications audio et vidéo à contraintes temporelles strictes, un contrôle de la gigue et du débit.

Pour mettre en place ces services il faut ajouter aux routeurs quatre fonctions principales qui sont :

- **L'ordonnement de paquets :** pour gérer la retransmission des différents paquets en utilisant un ensemble de files d'attente, chaque file correspondant à une classe de service.
- **La classification :** pour faire le contrôle de trafic, chaque paquet doit être réparti dans une classe parmi différentes classes, tous les paquets appartenant à une même classe seront traités de la même façon par l'ordonnancement de paquets.
- **Le contrôle d'admission :** un algorithme de décision permet au routeur de savoir s'il peut garantir la QoS demandée par un flux et ceci sans dommage pour la QoS des autres flux déjà existants.
- **RSVP :** Le protocole de réservation RSVP.

→ RSVP :

Pour faire des réservations pour les classes GS et CL, on a besoin d'un protocole de signalisation. L'IETF a spécifié le protocole RSVP. C'est un protocole de signalisation employé par les hôtes pour demander des qualités spécifiques de service pour des flux applicatifs particuliers. Les sources de trafic génèrent périodiquement des messages RSVP de type « Path » qui permettent aux récepteurs de connaître les caractéristiques du trafic des émetteurs. Les récepteurs répondent à ces messages par des messages « Resv » qui permettent aux nœuds de faire les réservations souhaitées par les émetteurs.

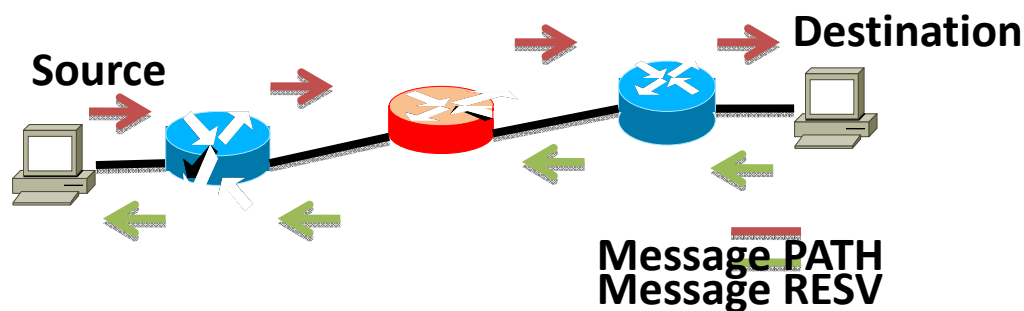


Figure III.1: Le protocole RSVP

III.1.2. Le modèle DiffServ

→ Principe du modèle DiffServ

Quand le groupe de travail du modèle DiffServ a été créé en 1997, il s'est concentré sur la définition d'un modèle simple, modulaire, dont la mise en œuvre peut s'effectuer d'une manière graduelle dans l'Internet existant. Le modèle a été défini par l'IETF dans le RFC 2475, son principe de base est la création de diverses classes de services fournissant chacune d'entre elles une qualité de service différente. Ces classes se distinguent les unes des autres par la présence d'un code dans le paquet IP : le DSCP (DiffServ Code Point). Le marquage des paquets selon la qualité de service qu'il nécessite, se fait en périphérie du réseau, soit à la source directement ou soit au niveau du routeur de bordure. Ensuite, les routeurs internes du réseau déterminent la priorité du paquet et le traitent en fonction de celle-ci. Chaque classe de service de qualité implique la création d'un contrat : le SLA (Services Level Agreement). Ce contrat précise les règles concernant le délai, la bande passante, le taux de disponibilité, la valeur du DSCP, la taille des tampons de la file d'attente pour cette classe dans les routeurs et le choix de la politique à utiliser en cas de non respect du SLA (rejet des paquets, abaissement de la priorité du paquet...).

→ Les avantages de la différenciation de service

La différenciation de services présente les avantages suivants :

- Pour un opérateur, les paquets qu'il reçoit marqués pour une certaine classe peuvent appartenir à plusieurs sources.
- La complexité du traitement est concentrée dans les routeurs aux frontières du réseau. Ils effectuent les opérations « complexes » de contrôle de la validité du contrat pour les différentes classes de trafic. Dans le cœur du réseau, le traitement est plus simple, ce qui autorise une transmission rapide des données.
- La tarification du service est plus simple, il suffit de définir les paramètres de contrôles de classes de service.
- il n'y a plus nécessité de maintenir un état des sources et des destinations dans les routeurs, d'où une meilleure scalability.
- De plus, afin d'éviter le problème de passage à l'échelle inhérent aux solutions IntServ, le choix a été fait de traiter un nombre limité d'agrégats (paquets IP n'appartenant pas nécessairement à un même flux) plutôt que des flux individuels.

➔ **Fonctionnement**

Dans le modèle DiffServ, il n'existe pas de signalisation dans le cœur du réseau, seule une étiquette dans l'en-tête, le DSCP (DiffServ Code Point), est utilisée pour assurer le traitement différencié [RFC2474]. Cette capacité permet aux routeurs de garder un mode de fonctionnement relativement simple qui n'affecte pas considérablement leur capacité d'acheminement. La complexité est reportée sur les routeurs de frontière, se trouvant à l'entrée ou à la sortie d'un domaine [RFC2475].

Ces équipements sont chargés de déterminer la valeur de l'étiquette de chaque paquet en fonction d'un contrat (SLA), du service demandé et des mécanismes de contrôle de trafic. Le DSCP occupe les six bits de poids fort du champ DS (DiffServ) de l'en-tête IPV4.

La figure 3.1.1 montre l'emplacement de ce champ. Dans IPv4, il s'appelle le champ ToS, l'IETF la redéfinit en champ DS (DiffServ) pour effectuer la classification. En IPv6, il se situe à la place du champ Traffic Class comme le montre la figure 3.1.2

0	4	8	16	31
Version	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time to live	Protocol	Header Checksum		
Source Address				
Destination Address				

(1)

0	4	12	16	31
Version	Traffic Class	Flow Label		
Payload Length			Next Header	Hop Limit
Source Address (16 octets)				
Destination Address (16 octets)				

(2)

(02).0	1	2	3	4	5	6	7
I	DSCP			CU			

(03)

Figure III.2: Le champ DiffServ en IP, (01) IPv4, (02) IPv6 (3) La forme du Champ DiffServ [RFC 2474].

La signification des différents champs est la suivante :

- Le champ **In** : indique si le paquet est « In/Out profile » (peut être éliminé ou non).
- Le champ **DSCP** (DiffServ Code Point) indique la classe du paquet,
- Le champ **CU** n'est pas utilisé pour l'instant.

Il est plus facile de comprendre l'utilité des divers éléments qui forment le modèle, un exemple dans la **figure X.2** montre une configuration de réseau simple. Un site émetteur, souhaitant que ses flux bénéficient d'un traitement différencié dans le réseau, établit un contrat de service avec son fournisseur. D'un côté, le fournisseur s'engage à fournir la qualité demandée par l'émetteur. De son côté, l'émetteur est conscient des limitations imposées par le contrat.

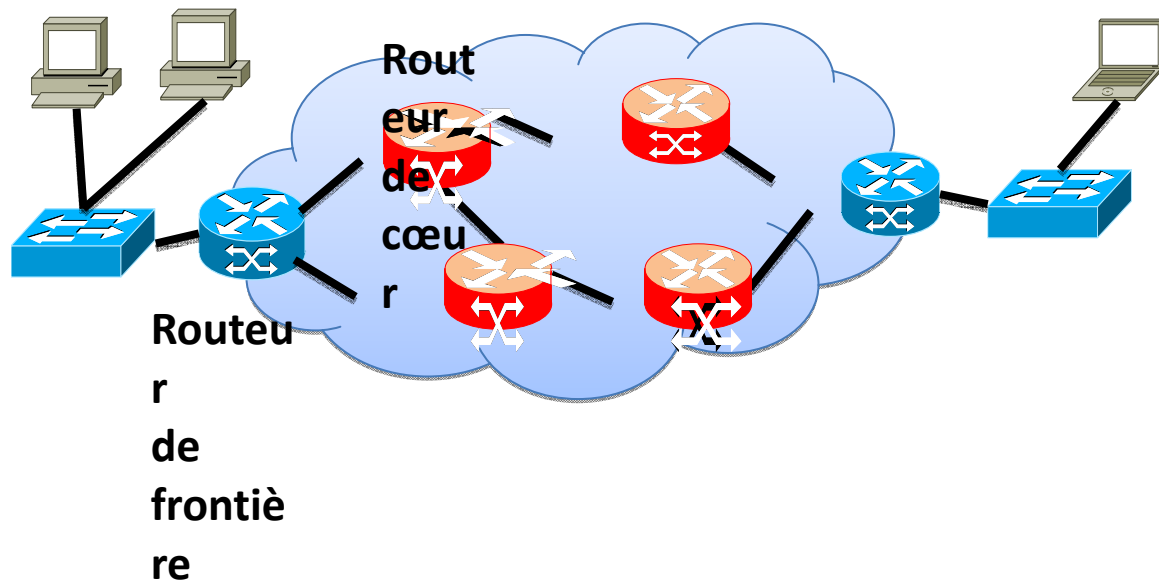


Figure III.3: Architecture Diffserv

- **Le client** peut effectuer un pré-marquage de ses flux avant de les transmettre au fournisseur. Signaler une préférence en termes de classe de service ou indiquer l'importance relative des paquets sont deux raisons qui justifient cette action. La priorité des paquets peut varier au sein d'un flux. Par contre, les principes de DiffServ exigent que tous les paquets d'un même flux appartiennent à une même classe de service afin d'éviter le déséquencement.
- **Le fournisseur** connaît les spécifications techniques du contrat établi avec chacun de ses clients. Quand le trafic produit par l'utilisateur arrive au routeur d'entrée du fournisseur, les flux sont identifiés et leur comportement est vérifié en fonction du contrat respectif. En cas de non-conformité, une action corrective est prise. Pour certains services, seuls les paquets conformes au contrat peuvent entrer dans le réseau. Pour d'autres services, tous les paquets sont acceptés, mais le routeur d'entrée modifie la priorité du paquet en fonction du niveau de conformité.

En quittant le routeur d'entrée, les paquets du site émetteur contiennent tous une étiquette, reflétant la classe de service souhaitée et le niveau de conformité du flux par rapport au contrat.

- **Les routeurs du cœur** du réseau ne modifient pas les étiquettes, ils se contentent de les utiliser pour choisir la file d'attente où le paquet est inséré. La file d'attente représente la classe de service. Les ressources des routeurs sont distribuées entre les files en fonction des services qu'ils supportent. Dans chaque file d'attente, un algorithme de rejet sélectif est utilisé pour éliminer en premier les paquets de basse priorité en cas de congestion.

La capacité des routeurs de sortie constitue une autre particularité de l'interconnexion entre deux domaines. Un routeur de sortie est le dernier routeur traversé par un paquet avant de quitter un domaine. Ces routeurs peuvent être chargés de traiter l'agrégat des flux quittant le domaine pour rendre le trafic conforme au contrat existant avec le prochain domaine.

Le dimensionnement de ressources est un facteur clé dans le maintien de la QoS. Les RFC2638 et RFC2748 sont deux propositions cherchant à implanter un certain type de signalisation entre l'utilisateur et le fournisseur. Dans ce cas, le contrat peut être modifié dynamiquement par l'utilisateur pour accroître ses capacités de transmission pour une durée limitée. Ce type de signalisation ne remet pas en cause la résistance du modèle au facteur d'échelle puisque des échanges ne s'effectuent qu'entre les clients et les routeurs d'entrée et ils n'ont pas besoin d'être connus par le reste du réseau.

➔ Routeurs d'entrée :

La structure d'un routeur d'entrée peut être décomposée en 4 modules comme le présente la figure X.3. Toute la complexité de l'architecture est concentrée dans ce type d'équipements. Ils doivent effectuer une classification qui peut aller jusqu'au niveau du micro-flux. Une fois le flux identifié, il est nécessaire de vérifier la conformité au contrat en utilisant un mécanisme adapté au service demandé. Ensuite une action est prise pour pénaliser les paquets non conformes. Finalement, avant d'injecter les paquets dans le réseau, la valeur du DSCP est actualisée en fonction du conditionnement subi.

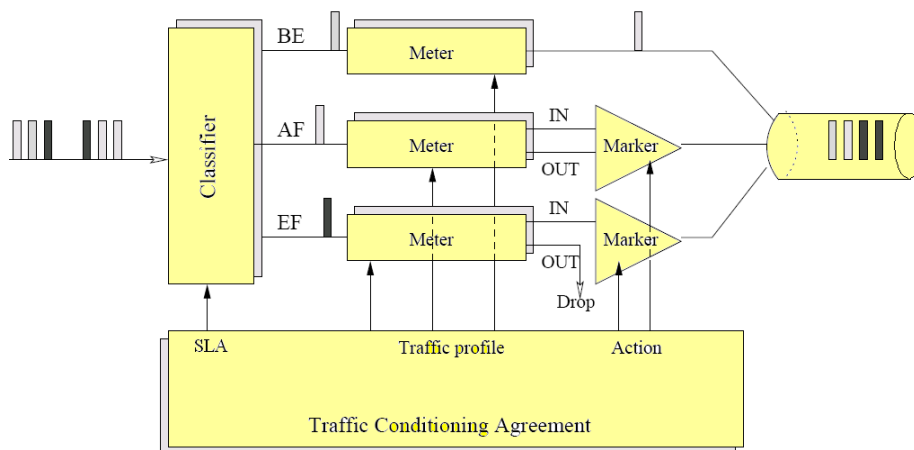


Figure III.4 : architecture interne d'un nœud interne.

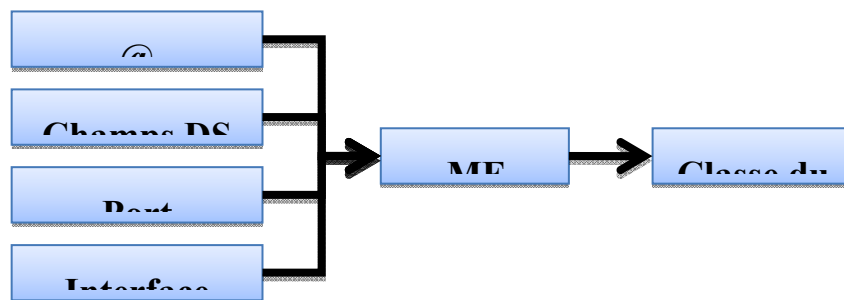
➤ La classification « Classifying »

Pour que les différents flux produits par les utilisateurs puissent être traités d'une manière différenciée, le routeur d'entrée doit tout d'abord effectuer une classification des paquets.

Elle se base toujours sur des champs de l'en-tête, mais le nombre des champs utilisés pour l'identification peut varier en fonction du contrat. Dans le modèle DiffServ, la classification peut aller jusqu'au niveau des micro-flux. La classification au niveau des micro-flux s'effectue dans les points où les flux entrent dans le réseau. Elle peut également avoir lieu dans un module de contrôle installé dans l'équipement de l'utilisateur. Dans ce cas, l'utilisateur est en charge de faire la mise en correspondance entre les applications et les classes de services. Le trafic est alors conditionné en tant qu'agrégat par le routeur d'entrée du fournisseur.

Les flux peuvent être classifiés selon deux types de **Classifieur** :

- **BA (BehaviourAggregate)** : classification établie uniquement en fonction de la valeur du champ DS;
- **MF (Multi-Field)** : classification établie selon la valeur d'un ou de plusieurs champs de l'en-tête du paquet (champ DS, adresse source, adresse destination, port...)



(1)



(2)

Figure III.5 : Les types de classification (1) La classification MF (2) La classification BA.

- **le conditionnement (policing) :**
 - **Vérification « metering »**

Un vérificateur est chargé de déterminer le niveau de conformité pour chaque paquet du flux arrivant dans le routeur. Cette valeur dépend du comportement

instantané du flux et des caractéristiques du contrat (SLA). Le nombre de niveaux varie en fonction du conditionnement requis par le service :

- Pour des services à bas délai, il est nécessaire que seuls les paquets conformes au contrat soient acceptés dans le réseau. Dans ce cas, une vérification à deux niveaux est suffisante : conforme ou non-conforme, IN ou OUT.
- Pour des services profitant de la capacité d'élimination sélective dans le cœur du réseau, un nombre supérieur à deux augmente la granularité de différenciation. Dans la définition du Comportement Assuré, 3 niveaux de priorité ont été définis : vert, orange et rouge.

- **Etiquetage « marking »**

Avant d'entrer dans le réseau, le champ DSCP de tous les paquets qui traversent le routeur d'entrée est mis à jour. La valeur de cette étiquette est le résultat des opérations de conditionnement. Il est possible d'identifier les deux informations qui établissent la différenciation : **la classe de service** et **la priorité du paquet** dans le champ DSCP.

Le DSCP qui forme l'étiquette DiffServ ne doit pas être modifié par les routeurs du cœur du réseau. Les routeurs de frontière peuvent quant à eux, modifier cette valeur lors des nouveaux conditionnements ou d'un problème de sémantique entre deux domaines voisins.

Les différentes combinaisons de vérificateurs et d'actions forment en grande partie un service. L'autre partie est formée par le choix du comportement dans le cœur du réseau et par la sélection d'une classe de service. Ce paramètre sera exploité dans les routeurs des cœurs du réseau. Etant donné que ces équipements ne doivent réaliser que des opérations simples et qu'ils doivent tous être conformes au standard, ce sont les modules du routeur d'entrée qui pourront plus facilement évoluer pour offrir des nouveaux services dans le futur.

- **Actions « dropping&shaping »**

L'action corrective à prendre pour les paquets non conformes d'un flux varie en fonction du service. Deux types de pénalisation peuvent être identifiés : l'élimination, la mise en forme et le marquage.

- ✓ **L'élimination (dropping)** est sans doute l'action la plus sévère, mais nécessaire au bon fonctionnement de certains services. Des flux interactifs contenant des contraintes temporelles fortes pourraient tirer profit de ce type d'action. Dans un service DiffServ interactif, les fournisseurs devront approvisionner leurs réseaux en fonction des contrats établis. L'entrée dans le réseau des paquets au-delà du contrat peut faire croître les files d'attente dans

les routeurs intermédiaires et provoquer une augmentation dans le délai de transmission qui rendrait l'interactivité impossible.

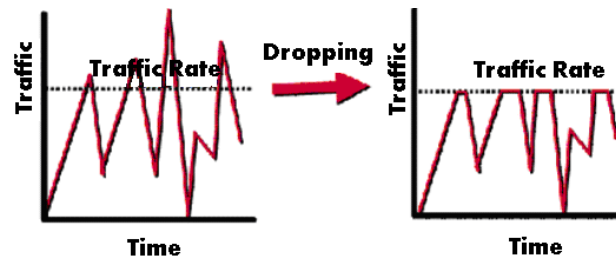


Figure III.6 : Le Dropping

- ✓ **La mise en forme (shaping)** consiste à retarder, si nécessaire, l'acheminement d'un flux pour le rendre conforme. Des services faisant appel à cette fonction peuvent être proposés aux clients qui n'ont pas les moyens de l'effectuer dans leur site. La mise en forme est une action très coûteuse en termes des ressources pour le routeur d'entrée, qui devra garder dans un tampon tous les paquets retardés jusqu'à ce qu'ils deviennent conformes.

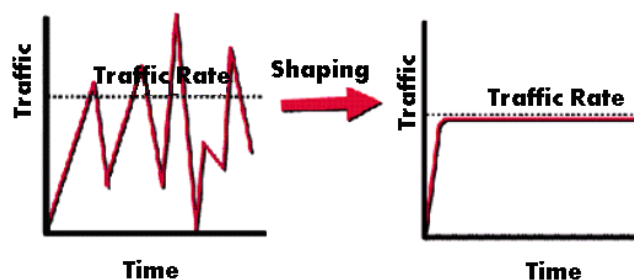


Figure III.7: Le Shaping

➔ **Routeurs du cœur du réseau**

A la suite des traitements effectués dans le routeur d'entrée, les paquets sont injectés dans le réseau. Les opérations propres aux routeurs du cœur du réseau sont relativement simples, mais celles-ci vont déterminer les performances du réseau, en termes de pertes et de délai, observés par les sites terminaux. L'un des objectifs du groupe DiffServ est de standardiser le traitement des paquets dans ces routeurs en fonction du champ DS. Tous les paquets portant une même valeur dans ce champ doivent être traités de la même manière. Le traitement différencié dans le cœur d'un réseau DiffServ implique deux axes : **les classes de service** et la **notion de priorités**. La figure X.4 montre l'architecture logique d'un routeur capable d'offrir ces deux niveaux de différenciation. Il est composé d'un classificateur, de plusieurs files d'attente, d'un mécanisme d'ordonnancement et d'un algorithme de gestion de file d'attente pour chaque file.

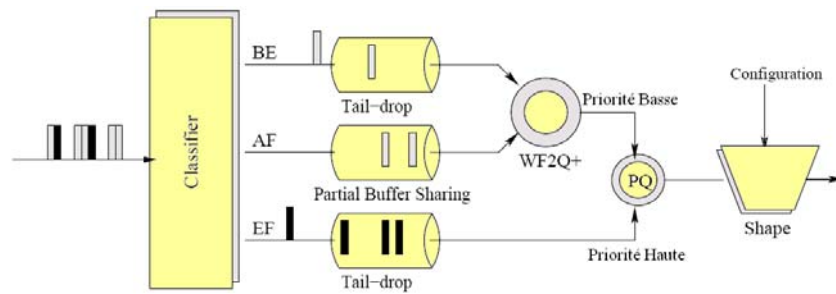


Figure III.8 Architecture du Routeur de cœur.

Le classificateur que l'on trouve dans ce type de routeurs réalise une classification beaucoup moins coûteuse que ceux qui se trouvent à l'entrée du domaine. Dans ce cas, les paquets sont différenciés uniquement en fonction de l'étiquette qu'ils portent, c'est-à-dire, de la valeur du champ DS attribuée par le routeur d'entrée. Une partie de cette étiquette sert à identifier la file d'attente sur laquelle le paquet doit être inséré, pendant qu'une autre partie de celle-ci peut être utilisée par l'algorithme de gestion de file d'attente pour améliorer la discrimination en cas de congestion. Chacune des files d'attente dans les routeurs représente une classe de service. Leurs propriétés, en termes de bande passante ou de retard moyen observé, dépendent du mécanisme d'ordonnancement.

Les contraintes des comportements à bas délai comme le Comportement Expédié, exigent une technique plutôt hybride, utilisant une combinaison de files prioritaires et des mécanismes de partage de ressources comme le WFQ ou le WRR.

A l'intérieur de chaque file, un mécanisme de gestion doit décider la manière dont les paquets seront éliminés en cas de congestion. Par exemple, pour une classe de service à bas délai, qui ne devraient jamais observer de pertes (théoriquement), la traditionnelle méthode FIFO (first in, first out) peut être utilisée. Par contre, pour les files correspondant aux classes AF [RFC2597], le standard demande la mise en œuvre d'une méthode capable de discriminer les paquets en fonction de leur précedence. Des méthodes comme WRED ou RIO peuvent être utilisées pour assurer cette fonction. Le champ DSCP exprime le comportement par nœud (PHB : Per-Hop Behavior) qui sera accordé au paquet. Un PHB est, d'après le [RFC2474], "la description des caractéristiques d'acheminement qui seront observés par tous les paquets comportant le même DSCP". L'utilisation d'un PHB ou d'un groupe de PHBs ajoutée aux opérations de conditionnement réalisées à l'entrée d'un domaine forment ce qu'on appelle **un service DiffServ**. Trois types de comportements ou PHBs ont été standardisés. Le seul PHB qui peut être considéré comme obligatoire est celui des sélecteurs de classe; il a été conçu pour garder une certaine compatibilité avec l'existant (le champ ToS). L'implantation d'autres comportements dans un réseau dépend des besoins du domaine. La seule contrainte pour le support d'un nouveau PHB exige que les PHBs supportés auparavant ne soit pas affectés par l'implantation de celui-ci. Il faut remarquer que, pour chaque PHB standardisé, les valeurs de DSCP

proposées ne sont que des recommandations. A l'intérieur d'un domaine, l'administrateur est libre de choisir les valeurs qui facilitent la mise en œuvre du modèle. Ceci demande pourtant, l'implantation des mécanismes de re-marquage à l'entrée du domaine pour faire la correspondance entre les DSCPs locaux et leurs équivalences dans les domaines voisins.

III.1.3. Les Comportements Standardisés

→ La classe Best Effort (priorité basse)

PHB par défaut et dont le DSCP vaut 000000, le principe du Best Effort se traduit par une simplification à l'extrême des équipements d'interconnexion. Quand la mémoire d'un routeur est saturée, les paquets sont rejetés. Le principe de bout en bout de l'Internet est aussi adopté pour le contrôle de flux grâce à différents algorithmes. Les principaux inconvénients de cette politique de contrôle de flux sont un trafic en dents de scie composé de phases où le débit augmente puis est réduit brutalement et une absence de garantie à long terme. Le **tableau X.1** présente le code DSCP recommandé pour le comportement BE.

DSCP	PHB
000 000	BE

Tableau III.2: DSCP associé à BE

→ Acheminement Expédié (Expedited Forwarding)

Les PHBs sont, par définition, des éléments indépendants qui peuvent être utilisés pour construire des services. Pour le moment, il n'existe aucune proposition explicite qui utilise les Sélecteurs de Classe ou l'Acheminement Assuré pour construire un service. Par contre, l'Acheminement Expédié est une exception puisqu'il a été conçu, dès le départ, comme étant le composant d'un service bien spécifique : **le service Premium**. Le service Premium ou câble virtuel [RFC2638] est destiné aux applications demandant un délai et une gigue très faible telle que le nécessite des applications comme la téléphonie sur IP. Il part du principe que le délai introduit par le réseau est dû à l'attente des paquets dans les files des routeurs. Il faut donc garantir que la taille de ces files soit presque nulle. Pour ceci il est nécessaire de mettre en œuvre des mécanismes d'administration qui assurent que, à tout moment, le débit d'entrée soit inférieur à la capacité de sortie.

L'Acheminement Expédié (EF pour ses sigles en anglais) est défini dans RFC2598. Pour qu'un routeur soit conforme à la définition de ce comportement, il doit assurer que le débit d'émission du trafic EF soit toujours supérieur ou égale à un certain seuil. Si ce seuil est connu et peut être configuré, les mécanismes de gestion externes peuvent faire respecter la relation entre le débit d'entrée et le débit de sortie dans tous les nœuds du domaine. Pour la mise en œuvre d'un PHB, plusieurs techniques peuvent être utilisées. Dans l'étude comparative faite dans la RFC 2598, il faut pourtant déduire que la meilleure option est l'utilisation d'une file prioritaire

réservée au trafic EF. Avec une telle architecture, le délai observé par les flux EF est réduit au maximum, mais une mauvaise administration peut entraîner la famine des autres classes de service. En conséquence, des mesures limitant le trafic EF doivent être prises dans chaque équipement qui implante le comportement. Le tableau X.2 présente le code DSCP recommandé pour le comportement EF.

DSCP	PHB
101 110	EF

Tableau III.3 : DSCP associé à EF.

➔ Acheminement Assuré (Assured Forwarding)

L'Acheminement Assuré est sans doute le comportement le plus général qui existe. Il est composé de plusieurs classes de service au sein desquelles un deuxième niveau de différenciation est introduit : les précédences. La précedence d'un paquet reflète sa priorité, c'est-à-dire, le dommage que peut entraîner sa perte. Au sein d'un routeur cette information peut être utilisée pour améliorer considérablement la fonction de rejet de paquets en cas de congestion. Suivant le même principe, la précedence attribuée aux paquets d'un flux peut être régie par un contrat de service. Plus le pourcentage des paquets marqués prioritaires sera important, moins le flux subira de pertes.

Le RFC2597 définit 4 classes comportant 3 niveaux de précedence chacune. L'Acheminement Assuré (AF en anglais) est donc composé par un groupe de 12 PHBs interdépendants. Ceci veut dire que l'implémentation d'un de ces PHBs est inutile sans l'implémentation des autres. Il a été décidé par l'IETF qu'aucune relation d'ordre ne devrait exister entre les classes. L'administrateur réseau est responsable du partage des ressources entre les classes en fonction de ses besoins. Par contre, il est nécessaire que tous les paquets d'un même micro-flux appartiennent à la même classe pour éviter le déséquencement des paquets. En ce qui concerne la précedence, le standard demande que, dans une même classe, les paquets de précedence X soient acheminés avec une probabilité supérieure à celle donnée aux paquets de précedence Y, si $X < Y$. On remarque la relation inverse qui existe entre la précedence et le niveau de priorité. Les PHBs qui forment le comportement AF sont identifiés par AFij, où i dénote la classe (1 à 4) et j dénote la précedence (1 à 3). La table suivante présente les codes DSCP proposés par le RFC2597 pour représenter les PHBs AF.

PHBsAFij	classe 1	classe 2	classe 3	classe 4
précedence 1	001 010	010 010	011 010	100 010
précedence 2	001 100	010 100	011 100	100 100
précedence 3	001 110	010 110	011 110	100 110

Tableau III.4 : DSCPs associés à AF

III.2. La diffusion consciente de DiffServ

La diffusion de la vidéo dans les réseaux à commutation de paquets peut être affectée par plusieurs facteurs, tels que la perte des paquets, le délai et la gigue. Plusieurs approches existent pour traiter ces problèmes. Une de ces approches est la diffusion de la vidéo consciente de DiffServ qui propose l'intégration entre un réseau DiffServ et l'application de diffusion de la vidéo.

III.2.1. Concept fondamental

La diffusion consciente de DiffServ est une approche pour la diffusion de la vidéo qui consiste à mettre en correspondance la sémantique des flux vidéo et les classes de trafic d'un réseau DiffServ. L'idée générale derrière ce concept est que la performance des systèmes peut être améliorée si les applications sont conscientes (c.à.d. utilisent) des capacités du réseau DiffServ. Les flux compressés générés par les codecs vidéo sont habituellement organisés dans une structure hiérarchique. Par exemple au niveau trame une vidéo compressée avec MPEG contient trois types de trames : **I**, **P** et **B**. La perte de chaque type de trame a un impact différent à la qualité visuelle du flux reçu. La perte d'une trame I affecte la qualité de façon plus importante que celle d'une trame B. Si une trame I est perdue, en plus de la perte de la trame en soi, quelques trames voisines peuvent être considérées aussi comme perdues dans la pratique puisqu'elles ne peuvent pas être décodées correctement malgré leur réception car l'information de référence de la trame I n'est pas disponible. On parle alors de **propagation d'erreurs**.

Quand on parle de diffusion consciente de DiffServ, on se réfère aux méthodes où le flux vidéo est, d'une façon ou d'une autre, séparé dans des parties (couches, trames, tranches, etc.) qui sont marquées et acheminées dans un réseau DiffServ par des classes ou des priorités de rejet différentes. Par conséquent le cas où un flux vidéo en entier est isolé et transporté dans une seule classe DiffServ (une classe premium, par exemple) n'est pas considérée comme diffusion consciente de DiffServ.

➔ Application consciente de DiffServ

Une application de diffusion de la vidéo consciente de DiffServ marque les paquets qu'elle émet en fonction du type d'information vidéo codée qu'ils contiennent. Ainsi, le traitement différencié dans le réseau DiffServ devrait offrir une meilleure qualité visuelle au moyen d'assurer le débit, limiter la gigue ou bien rejeter d'abord les paquets moins importants en cas de congestion. Ce type de marquage est appelé **marquage sémantique**.

➔ Correspondances sémantiques et classes de service DiffServ

- **Correspondance directe entre les types des trames et les niveaux de rejets AF**

Dans cette approche le flux vidéo est mis dans une seule classe AF et le marquage est effectué comme suite :

- Les paquets des trames I sont marqués comme des paquets verts (green)
- Les paquets des trames P sont marqués comme des paquets jaunes (yellow)
- Les paquets des trames B sont marqués comme des paquets rouges (red)

Au niveau du routeur, un algorithme de gestion de file active multi-priorités comme RIO est l'élément clé pour implémenter le comportement AF qui fournit un rejet différencié des paquets dans une situation de congestion. Avec RIO, les paquets rouges sont rejetés avant les paquets jaunes, qui sont rejetés avant les paquets verts. Comme cela les paquets des trames I sont protégés et la propagation des erreurs est réduite (ou éliminée). L'avantage de cette approche est sa simplicité.

➤ **Utilisation des flux en couches et multiple classe DiffServ**

Un flux vidéo scalable est un flux décomposé en deux ou plusieurs couches. Il y a une couche de base et une ou plusieurs couches d'amélioration. La réception de la couche de base seule donne une qualité acceptable, mais plus on reçoit des couches d'amélioration plus la qualité est accrue. En partant de ce principe les couches d'un flux scalable sont associées aux classes de DiffServ. En général chaque couche est assignée à une classe DiffServ spécifique.

Par exemple, on associe la couche de base à la classe EF, comme ça le débit, le délai et la gigue peuvent être garantis. Également, les couches d'amélioration peuvent être associés aux différents niveaux de priorité d'une class AF. Cette Approche fournit une discrimination granulaire plus fine. Mais le fait de diviser le flux en plusieurs flux peut entrainer des problèmes de désordre et de synchronisation.

➤ **Classification basé sur l'estimation de distorsion**

Le concept général de cette approche est le suivant :

Avant de transmettre chaque paquet du flux vidéo, l'émetteur estime l'impact de la perte de chaque paquet sur la qualité et lui attribue une marque DiffServ selon cette estimation, l'émetteur simule la perte de chaque paquet et décode la trame corrompu, ensuite cette trame est comparé avec la trame original en utilisant une technique d'évaluation objective tel que le PSNR, le résultat est utilisé pour le marquage DiffServ.

1.5.1 Conclusion

La QoS permet d'utiliser l'infrastructure réseau préexistante sans surdimensionné le réseau et engendrer des dépenses supplémentaires. Afin de garantir

une qualité de service, il est important de différencier plusieurs types de trafic et de ne pas offrir les mêmes performances et services pour chaque application. D'autre part, la Qualité de Service s'applique sur tout le réseau. L'objectif est d'avoir une qualité autant en périphérie des réseaux (entreprise) qu'en communication inter réseaux (Internet Service Provider ou opérateur). Pour que la QoS soit optimale, elle doit être mise en place de bout en bout. Les applications en Temps Réels comme la voix sur IP ou la vidéo requièrent des délais de transmission faible, de gigue et des exigences en termes de bande passante.

CHAPITRE IV

APPLICATIONS ET RESULTATS PRATIQUES

IV.1. Simulation

Les réseaux de télécommunications, et en particulier les réseaux informatiques connaissent une expansion sans précédent. Devant l'évolution des techniques et de la technologie, de nombreuses solutions sont envisageables pour un même problème. La simulation permet ainsi de tester sans aucun coût ces nouvelles technologies, les nouveaux protocoles mais aussi d'anticiper les problèmes qui pourront se poser dans le futur.

IV.2. Présentation de NS-2 :

IV.2.1. Développement du simulateur NS

Le simulateur de réseaux NS Network Simulator, est un environnement de simulation à base d'évènements discrets, il offre un support substantiel pour la simulation des protocoles de contrôle des transmissions (TCP), des protocoles de routage, et des protocoles de multi-casting à travers des réseaux filaires et sans fil (réseaux locaux et satellitaires). Au départ, NS était une variante de «The REAL Network Simulator» en 1989, puis, a évolué principalement au cours de ces dernières années grâce à des efforts administrés par l'université de Berkeley. En 1995, le développement de NS a été soutenu par le DARPA (Defense Advanced Research Projects Agency), et cela, à travers le projet VINT (Virtual InterNet Testbed). Actuellement, NS est soutenu par le DARPA avec SAMAN et par NSF avec CONSER tout en collaborant avec d'autres organismes de recherches tels que ACIRI.

IV.2.2. Le simulateur NS-2

Le simulateur de réseaux NS-2 est un simulateur orienté objet, développé sur la base du langage C++ et du langage de script OTcl (Object Tool command language). La combinaison de ces deux langages permet, d'une part, la définition et la création des objets manipulés, ce qui est réalisé efficacement (et rapidement) par le langage C++, et d'autre part, la configuration des objets et la gestion des évènements à travers le langage de script OTcl. L'interpréteur OTcl, dérivé du langage Tcl, permet également une interaction avec les utilisateurs en leur permettant d'écrire des scripts pour la définition du modèle de réseau (le nombre de nœuds, les liens, ...etc.), la détermination du trafic dans le réseau (sources, destinations, type de trafic, ...etc.), et la définition des protocoles utilisés, et cela, sur la base du langage Tcl. Le résultat d'une simulation est un fichier trace d'extension «.tr», contenant tous les évènements de la simulation. Un traitement ultérieur de ce fichier permet d'en soustraire l'information souhaitée. Par ailleurs, le simulateur permet la création d'un fichier d'animation (d'extension .nam), permettant de visualiser la simulation sur l'interface graphique NAM (Network Animator). La figure suivante présente de façon simplifiée, le point de vue d'un utilisateur de NS.

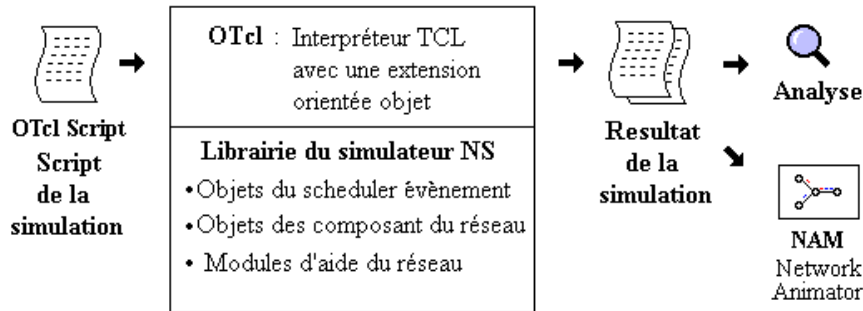


Figure 4.1: Schéma de l'utilisation de NS

Le tableau suivant résume les différentes versions des logiciels utilisés dans ce mémoire :

Outil	Version	Description
Back Track	3 Final	Système d'exploitation Linux
Ns-2	2.30	Simulateur Réseaux
nam	1.12	Animateur réseau
Xgraph	12.1	Traceur des graphes

IV.2.3. Développement de nouveaux composants

➔ Principes de base

NS est un simulateur à événements discrets orienté objet. Il est écrit en C++ avec une interface textuelle (ou shell) qui utilise le langage OTcl (Object Tool Command Language). L'OTcl est une extension objet au langage de commande Tcl. Le langage C++ sert à décrire le fonctionnement interne des composants de la simulation. Pour reprendre la terminologie objet, il sert à définir les classes.

Quant au langage OTcl, il fournit un moyen flexible et puissant de contrôle de la simulation comme le déclenchement d'événements, la configuration du réseau, la collecte de statistiques, etc. L'application NS se compose de deux éléments fonctionnels: un interpréteur et un moteur de simulation. Au moyen de l'interpréteur l'utilisateur est capable de créer le modèle de simulation ce qui revient à assembler les différents composants nécessaires à l'étude. Les composants du modèle de simulation sont appelés objets ou encore instances de classe. Le moteur de simulation effectue les calculs applicables au modèle préalablement construit par l'utilisateur via l'interpréteur. NS bénéficie de toutes les possibilités qu'offrent les techniques objets comme l'héritage, le polymorphisme, la surcharge, etc. L'héritage permet d'élaborer des arborescences de classes. Le modèle de simulation est construit à partir d'une arborescence de classes qui en fait se dédouble:

- une définie en OTcl dite arborescence interprétée. Elle est utilisée par l'interpréteur et est visible par l'utilisateur.
- une définie en C++ que l'on nommera compilée. Elle forme l'arborescence utilisée par le moteur de simulation.

Les deux arborescences sont très proches l'une de l'autre. Du point de vue de l'utilisateur, il y a une correspondance univoque entre une classe d'une arborescence et une classe de l'autre arborescence. La figure 1 montre la dualité des classes qui peut exister dans NS.

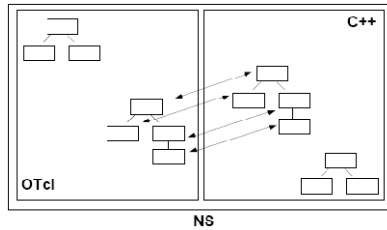


Figure 1: Dualité des classes OTcl et C++

Le principe général de la création des objets du modèle de simulation est le suivant: l'utilisateur crée un nouvel objet via l'interpréteur OTcl. Le nouvel objet interprété est cloné en un objet compilé correspondant dans le simulateur.

➔ **1.1. Arborescence des classes compilées**

La figure 2 représente l'arborescence de classes utilisée par le simulateur. Les classes visibles au niveau de l'interpréteur comportent une déclaration dans la classe TclClass. Nous verrons également l'objectif des différentes classes. Le nom des classes correspond à celui utilisé dans le code source.

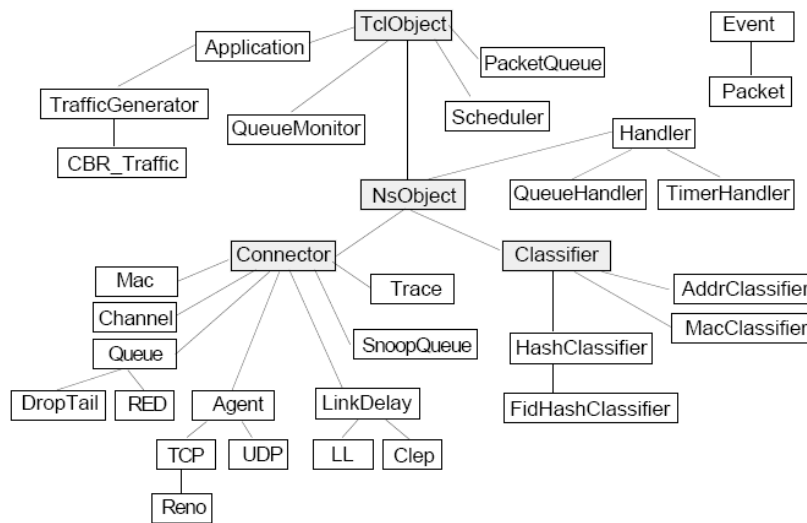


Figure 2: Arborescence de dérivation des classes C++ du simulateur

IV.2.4. Architecture du réseau

➔ **Nœud (node)**

La classe node est une classe OTcl: elle n'a donc pas d'existence en tant que telle dans le simulateur. Cette classe et ses méthodes sont définies dans le fichier tcl/lib/ns-node.tcl. Un nœud est une collection de classifieurs et d'agents. Le classifieur dé-multiplexe les paquets. L'agent est habituellement l'entité d'un protocole. L'assemblage des composants est représenté par la figure 6:

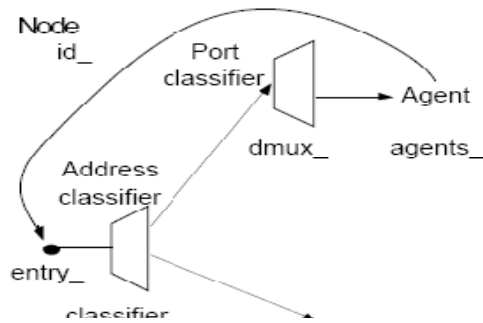


Figure : Composants d'un nœud

➔ 3.2. Lien

Le lien sert à relier les nœuds. Il modélise le système de transmission. Le lien est principalement caractérisé par un délai de propagation et une bande passante. C'est une classe OTcl qui regroupe un ensemble de composants dérivés de la classe Connector. Cette classe et ses méthodes sont définies dans le fichier `tcl/lib/ns-link.tcl`. Des liens plus sophistiqués peuvent être dérivés de cette classe. Quelque soit le type du lien, il comporte 5 références sur des objets qui le composent. Le lien peut se représenter selon la figure 7.

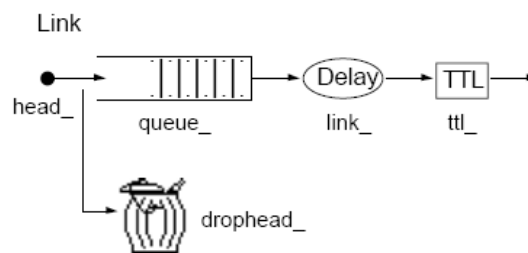


Figure 7: Composants d'un lien.

Les 5 instances variables suivantes ont la sémantique:

- `head_`, point d'entrée du lien, il pointe sur le premier objet du lien.
- `queue_`, référence la file d'attente de paquets. Cette file modélise celle que l'on trouve habituellement aux interfaces de sortie.
- `link_`, référence l'élément qui caractérise le lien souvent en terme de délai et bande passante.
- `ttl_`, référence l'élément qui manipule la durée de vie de chaque paquet.
- `drophead_`, référence l'élément qui traite les pertes de paquets au niveau de la file d'attente.

➔ 3.3. Agent

L'agent est un autre composant d'un nœud. Il modélise les constructeurs et les consommateurs de paquets IP. La classe agent fournit des méthodes utiles au développement de la couche transport et à d'autres protocoles du plan de signalisation ou de gestion. Cette classe est à la fois dans l'interpréteur et dans le simulateur. C'est la classe de base pour définir des nouveaux protocoles dans NS. Elle fournit l'adresse locale et de destination, les fonctions pour générer les paquets, l'interface à la classe

application. Actuellement NS comporte de nombreux agents citons: UDP, protocoles de routage, différentes versions de TCP, RTP, etc. Les principales commandes sont celles qui ont trait à des attachements:

- `$ns_ attach-agent $node_ $agent_` attache l'agent au nœud.
- `$ns_ connect $agent1 $agent2` établit une connexion entre les deux agents.
- `$agent_ attach-source $source_` attache une source de trafic à un agent.

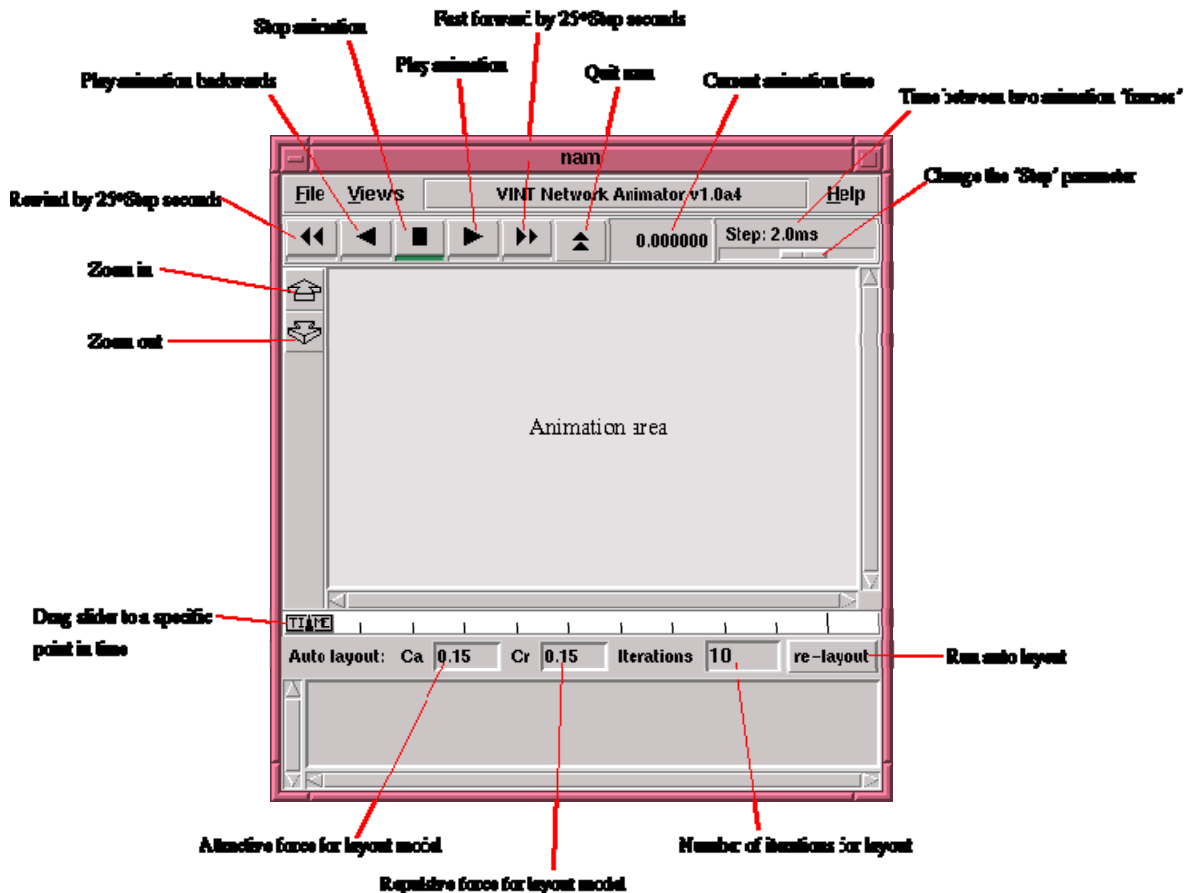
IV.2.5. 5. Interprétation des résultats

NS fournit deux moyens pour extraire des données de la simulation:

- La *trace* enregistre dans un fichier les changements d'états d'un paquet ou de valeur d'une variable.
- Le *moniteur* est un objet pouvant faire des calculs sur différentes grandeurs tel que le nombre de paquets ou d'octets arrivés, etc.

➔ L'interprétation à l'aide du NAM

Le logiciel NAM permet de visualiser une animation des paquets dans un réseau. La figure suivante donne une vue générale du NAM



Nous allons écrire un programme simple, qui va montrer maintenant comment en construisant un réseau de quatre nœuds avec deux sources de paquets. Lors de la

visualisation, on souhaite que les paquets d'une source apparaissent en bleu, les autres en rouge. Voici le texte de ce nouveau programme et la visualisation qui en est obtenue. On remarque la visualisation de la file d'attente de messages qui se forme au nœud 2, puisque les trafics en entrée sont assez proches de la saturation.

NB. Les commentaires (introduits par le caractère #) expliquent le rôle de chaque instruction ou partie de programme.

```
# Création d'un simulateur
set ns [new Simulator]
setnf [open out.nam w]
$ns namtrace-all $nf

# Lorsque la simulation sera terminée.
proc finish {}
{
    globalns nf
    $ns flush-trace
    close $nf
    execnamout.nam&
    exit 0
}

# Création de quatre nœuds
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

# Création de lignes de communication full duplex entre les nœuds
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n3 $n2 1Mb 10ms DropTail

# Création d'agents générateurs de paquets à vitesse constante
set cbr0 [new Agent/CBR]
$ns attach-agent $n0 $cbr0
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 set fid_ 1
set cbr1 [new Agent/CBR]
$ns attach-agent $n1 $cbr1
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 set fid_ 2

# création d'un agent vide, destiné à recevoir les paquets
set null0 [new Agent/Null]
$ns attach-agent $n3 $null0

# routage des trafics
$ns connect $cbr0 $null0
$ns connect $cbr1 $null0

# surveillance d'une file d'attente du noeud 2 vers le noeud 3
$ns duplex-link-op $n2 $n3 queuePos 0.5

# scénario de début et de fin de génération des paquets par cbr0
$ns at 0.5 "$cbr0 start"
$ns at 1.0 "$cbr1 start"
$ns at 4.0 "$cbr1 stop"
$ns at 4.5 "$cbr0 stop"

# Paramétrage des couleurs d'affichage
$ns color 1 Blue
$ns color 2 Red

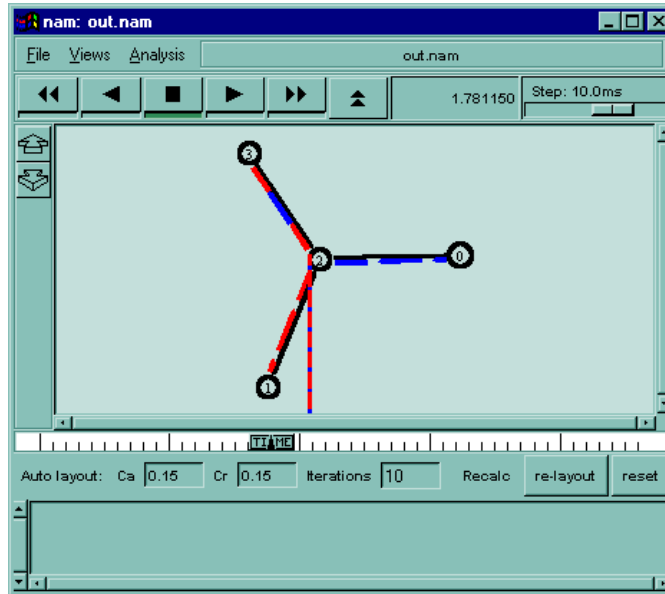
# la simulation va durer 5 secondes de temps simulé
$ns at 5.0 "finish"

# début de la simulation
```

```
$ns run
```

En exécutant ce programme (voir figure, on verra que la gestion de la file d'attente à l'entrée du lien 2 vers 3 n'est pas très équitable. Pour modifier cela, il suffit d'indiquer une gestion politique différente à cette file en modifiant la déclaration du lien en question par :

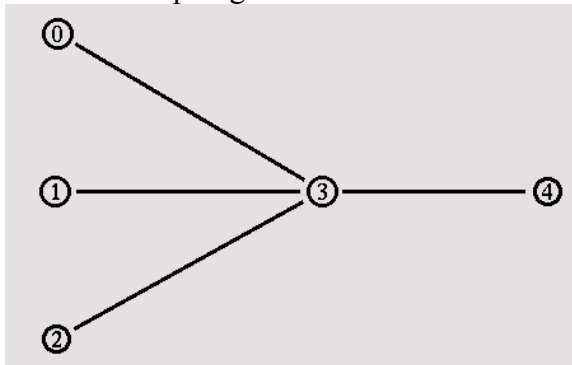
```
$ns duplex-link $n3 $n2 1Mb 10ms SFQ
```



➔ L'interprétation à l'aide du Xgraph

Parmi les outils utilisés par NS le 'xgraph' qui permet de créer des fichiers graphiques représentant les résultats de la simulation d'un réseau et pour mieux comprendre l'utilité de ce logiciel voici un exemple d'une TCL script.

- Tous d'abord on va créer la topologie suivante :



- Puis on va créer et sauvegarder les résultats dans des fichiers à l'aide des commandes suivantes :

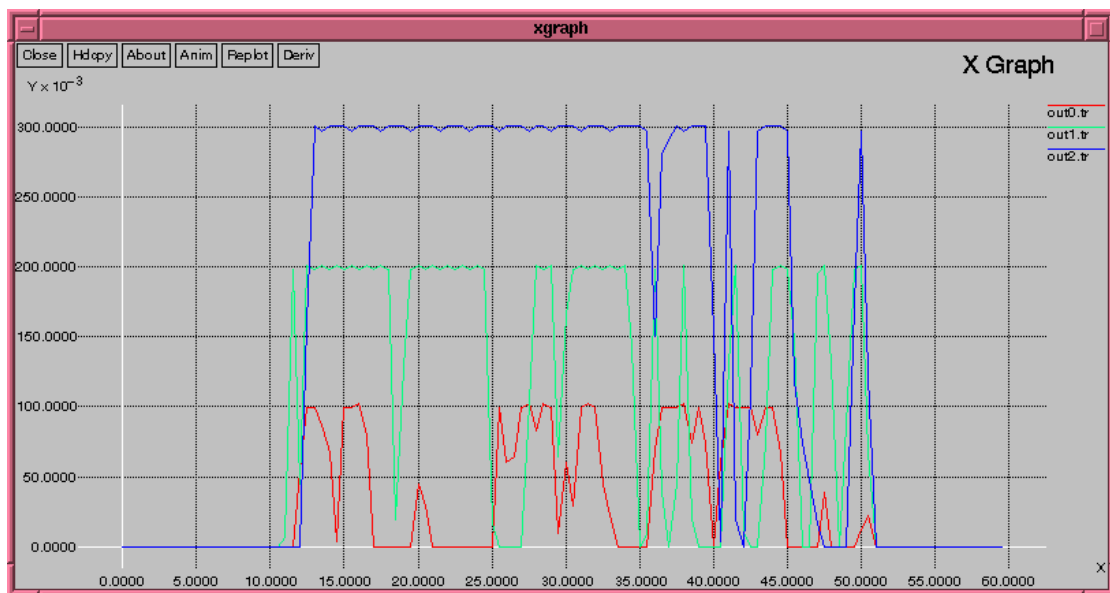
```
set f0 [open out0.tr w]
set f1 [open out1.tr w]
set f2 [open out2.tr w]
```

- Configuration du temps d'exécution de programme et l'exécution du programme

```
$ns at 0.0 "record"
$ns at 10.0 "$source0 start"
$ns at 10.0 "$source1 start"
$ns at 10.0 "$source2 start"
$ns at 50.0 "$source0 stop"
$ns at 50.0 "$source1 stop"
$ns at 50.0 "$source2 stop"
$ns at 60.0 "finish"

$ns run
```

- L'apparition du logiciel xgraph automatiquement et visualisation des resultants



IV.3. Architecture de la Simulation:

La figure suivante donne la structure générale de la transmission d'une vidéo en passant par l'enregistrement, le codage et la compression de la vidéo brute (dans notre cas vidéo de type YUV) au vidéo MPEG, packetization, le marquage des paquets, la transmission sur le réseau, la gestion de la file 'attente, le décodage et la correction de la vidéo et enfin l'évaluation de la qualité par le calcul de PSNR en déduisant la valeur de MOS.

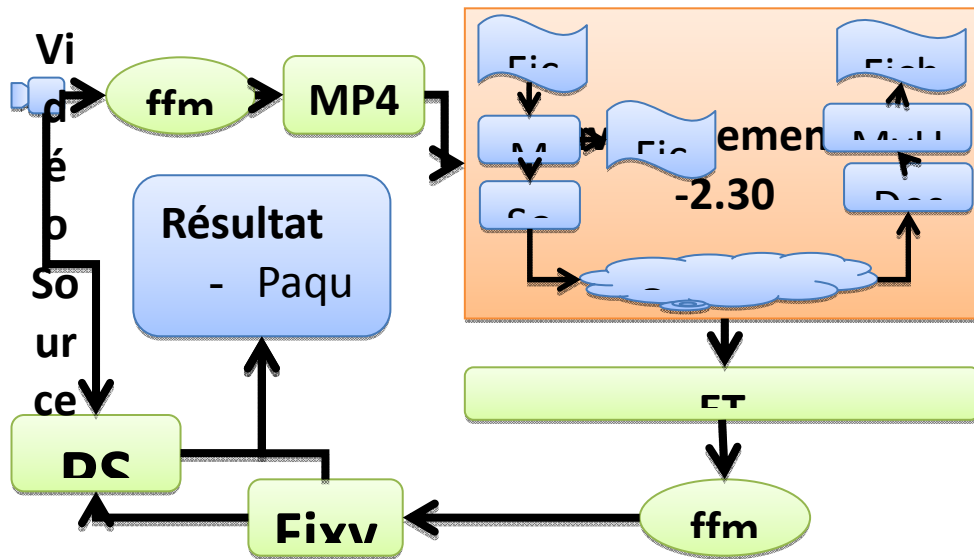


Figure: architecture de la Simulation

IV.3.1. Outils développés :

Pour pouvoir accomplir la simulation, nous avons développé des outils qui nous permettent le codage, le décodage, la packtization de la vidéo et le calcul de PSNR. Le tableau suivant résume ces outils.

Outil	Fonction
ffmpeg	Codec Multimédia. [22]
MP4	Outil utilisé pour la détection des trame I, P, B et H du fichier MPEG-4 de type m4v et la fragmentation des trames en des paquets prédéfinie par l'utilisateur (la taille utilisée est de 1000 octet).
Et	Générateur de la vidéo compressée reçue de type m4v.
PSNR	Calculateur de PSNR
Fixyuv	Correcteur de la vidéo reçue.
YUVviewer	Lecteur de vidéos non compressés

Tableau 7.1 : Outils développés.

IV.3.2. Scénarios simulés :

Dans le premier scenario aucune différentiation n'est effectuée entre les paquets de trafic de charge et ceux de la vidéos. Tandis que dans les deux derniers scénarios les paquets vidéo sont prioritaires, et même entre ces paquets il y a une différentiation. Les paquets des trames I sont prioritaires par rapport aux paquets des trames P qui sont prioritaires par rapport aux paquets des trames B.

IV.3.3. Etapes de la simulation :

➔ Encodage de la vidéo :

La conversion de la vidéo brute YUV vers une vidéo compressée m4v se fait grâce à outil FFmpeg CONVERTER.

FFmpeg est une open source fonction sous linux permet de convertir toutes les formats audio et vidéo en une seule ligne de commande, la forme de la commande est la suivante :

```
root#>ffmpeg [[infile options][`-i' infile]]... {[outfile options] outfile}...
```

Le tableau suivant donne quelques options utilisées dans ce mémoire :

Option	Description
-i	Donner le chemin de la vidéo à convertir.
-formats	Indique les formats disponibles, codec, protocole,...
-s size	entrez la résolution de la trame (frame), le format est de type 'WxH' (default 160x128). Ou les abréviations suivantes: <ul style="list-style-type: none"> • « qcif » 176x144 • « cif » 352x288 ` • « 4cif » 704x576 • « vga » 640x480 • « svga » 800x600 •
-r fps	Taux de trame par second (25fps par défaut).
-g size	Donner la longueur d'un GOP.
-intra	Utiliser seulement la trame Intra (I).
-bf frames	Utiliser les trames B supporté par MPEG 1,2 et 4.

Dans ce chapitre final, on a choisi comme séquence vidéo pour le test; quatre séquences connues dans ce domaine de recherche, de format YUV, la première est « Stefan », la deuxième est « Bus », la troisième est « Foreman » et la quatrième « Hall_cif », les vidéo se sont d'une format CIF « 352x288 ».

Dans ce mémoire on utilise une compression MPEG-4 telle que la taille du GOP est de 12 avec trois types de compression ou nous utilisons:

- Seulement des trames I.
- Les trames I et P.
- Les trames I, P et B avec M=2.

Le tableau suivant indique les propriétés des vidéos utilisées :





	Nom	Nombre de trames	Résolution	Taille de la vidéo brute	Taille de la vidéo compressée		
					Seulement les trames I	I et P	I, P et B
	Stefan_cif	90	352x288	13.1 Mo	684,9 Ko	212,4 Ko	389,3 Ko
	Bus_cif	150	352x288	21,8	918,5 Ko	309,2 Ko	475,2 Ko
	Foreman_cif	300	352x288	43,5 Mo	1 Mo	377,9 Ko	365,3 Ko
	Hall_cif	300	352x288	43,5 Mo	1,1 Mo	414,8 Ko	385,2 Ko

Tableau: Encodage de la vidéo

Le rapport de compression est compris entre 20:1 et 140:1

➔ **Packetization et détection des trames**

MP4 est un outil développé en C++, qui permet de détecter les différentes trames I, P, B et H ([Header] entête de fichier MPEG4) et la taille de chaque trames en suite il va envoyer dans le réseau en spécifiant l'adresse IP, le port, la taille de paquet et le chemin de la vidéo compressée en MPEG-4. L'exécution de cet outil se fait dans le Shell de commande selon la forme suivante :

```
root#>mp4 -send <ip><port><tp>fichier>ST
```

- Ip Adresse IP
- Port Port udp
- Tp Taille du paquet
- Fichier Le fichier a envoyé

Cet outil génère un fichier trace **ST** qui contient des informations nécessaire pour la différentiation entre les trame I, P et B.

Voici un extrait du fichier **ST** :

```
0 H 29 1 segmat 33 ms
1 I 3036 4 segmat 66 ms
2 P 659 1 segmat 99 ms
3 B 357 1 segmat 132
```

➔ Simulation avec NS-2 :

Dans ce mémoire on va utiliser une architecture simple constitué de deux routeurs de frontière, un routeur de cœur, un serveur vidéo, un générateur de trafic de charge (pour la congestion du réseau), des stations réceptrices, La topologie montrée dans la figure suivante est utilisée dans trois scenarios.

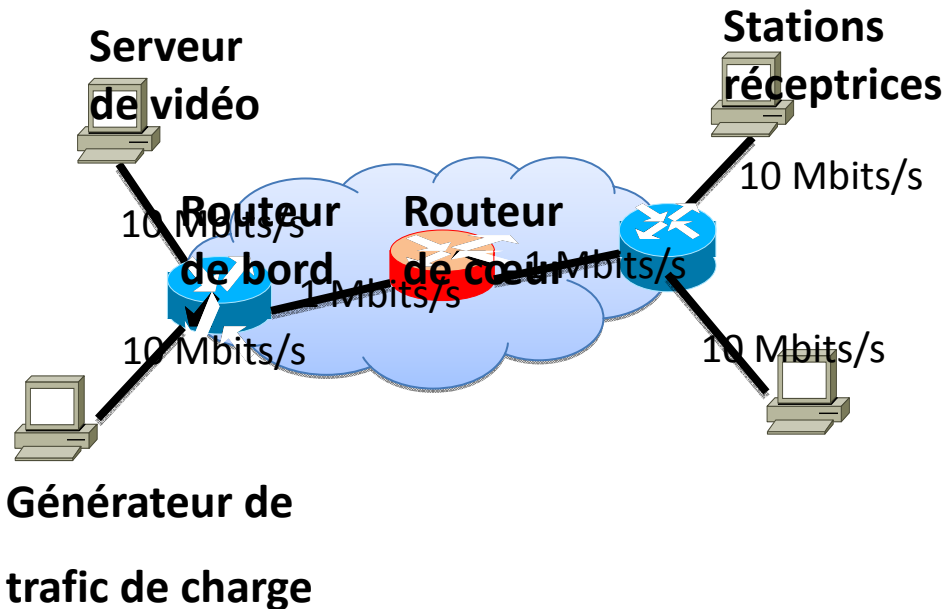


Figure: architecture du réseau simulé.

IV.3.3..1. Simulation du scenario Best Effort :

Dans le scenario Best Effort, il n'y a aucune différenciation entre les paquets de la trame I, P, B, les paquets de type élastiques et les paquets de la voix. L'exécution de Scenario Best Effort dans le Shell qui va afficher les différents paramètres tels que la durée de la simulation, Nombre du paquet émis, nombre du paquet reçu et le nombre des paquets perdu. Tous les résultats sont enregistrés dans des fichiers (BestEffort.nam, emis_be, recus_be) avec une possibilité de visualisé une animation résultats avec l'outil NAM.

- **emis_besteffort:** contient le temps d'émission, le numéro et la taille de chaque paquet émis.
- **recus_besteffort:** contient le temps de réception, le numéro et la taille de chaque paquet reçu par le client vidéo.

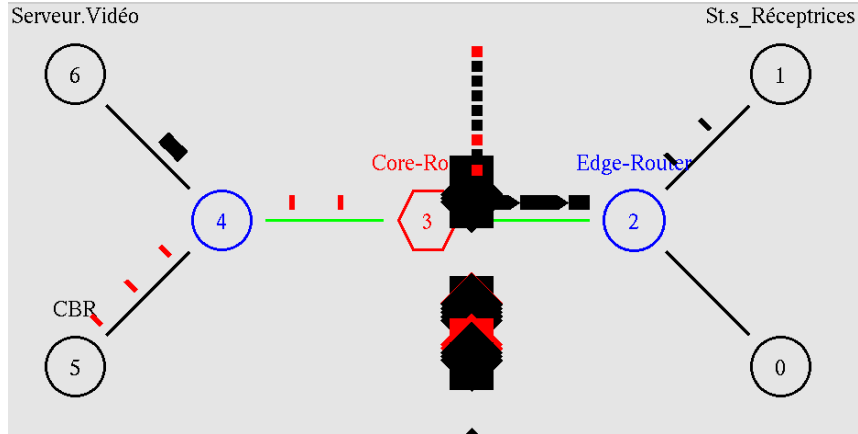


Figure : Visualisation des résultats avec l’outil Nam

IV.3.3..2. Simulation du scénario DiffServ:

Dans ce scénario les paquets vidéo sont prioritaires, et même entre ces paquets il y a une différenciation. Les paquets des trames I sont prioritaires par rapport aux paquets des trames P qui sont prioritaires par rapport aux paquets des trames B. L’exécution de Scenario DiffServ dans le Shell qui va afficher les différents paramètres tels que la durée de la simulation, Nombre du paquet émis, nombre du paquet reçu et le nombre des paquets perdu (voire la figure suivante), comme on peut les visualisé l’animation dans NAM (Network Animator).

```

Shell - Konsole <3>
-----
République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique
Département d'électroniques
Option Signal & Communication
-----
Mémoire de magistère:
-----
Optimisation de la transmissions d'un flux Multimédia MPEG-4 dans un réseau IP
par la methode DiffServ
-----
Préparé par:
      KADIK Rachid
      Ingénieur en Télécommunications de l'INT-TIC d'Oran.
-----
la durée de la Simulation est : 10.433333333333334 secondes.
Sauvgarde de 311 paquets

Statistiques des Paquets
-----
      Total      Reçus      perdues      Rejetés
-----
All      12895      1317      10945      633
0        12421      1109      10690      622
I         71        62         9         0
P        147        127        20         0
B         256        19         226        11
-----
Fin de la Simmulation
Email: rachid_kadik@hotmail.com
Exucution du nam...
    
```

Figure: Visualisation des résultats DiffServ dans le Shell de commande.

➔ Génération de la vidéo reçue

La génération de la vidéo reçue est faite à l'aide de l'outil gen.

gen est un outil développé en C++ est exécuté au Shell de linux par commande suivante :

```
root#>gen <fe><fr><ST><vs><vout>
```

Telle que

- Fe Fichier émis
- Fr Fichier reçus
- St Fichier référence
- Vs Vidéo source (m4v)
- vout Vidéo généré (m4v)

➔ Décodage de la vidéo reçue

Le décodage de la vidéo de format m4v au format YUV se fait à l'aide de la commande suivante :

```
root#>ffmpeg -s cif -g 12 -r 30 -i <fs> -r 30 <fd>
```

Telle que

- Fs Fichier source.
- Fd Fichier décodé

➔ Correction de la vidéo reçue décodée :

La vidéo décompressée reçue contient des trames perdues, pour le calcul de PSNR il faut que la vidéo source et la vidéo reçue contiennent le même nombre de trames, pour cela l'outil fixyuvremplace les trames perdues par la dernière trame bien reçue. Cor est outil développé en C++ et on peut l'exécuté au shell Linux à l'aide de la commande suivante :

```
root#>cor<sd><rd><st><in.yuv><out.yuv>
```

- Sd fichier emis
- Rd fichier reçu
- St fichier référence
- In.yuv vidéo éroné
- Out.yuv vidéo corrigé

➔ Calcul du PSNR

L'outil PSNR permet de calculé le PSNR entre deux vidéo et enregistré les valeurs de chaque trame dans un fichier.

```
root#>psnr<x><y><vs><vd>
```

- x La largeur de la vidéo
- y La hauteur de la vidéo
- vs Vidéo source
- vd Vidéo distordé


➔ Calcul de MOS

L'outil MOS permet de calculé le MOS entre deux vidéo et enregistré les valeurs de chaque trame dans un fichier.


```
Root#>mos<x><y><vs><vd>
```

IV.4. Résultats de la simulation


Les tableaux suivants résumant le rapport PSNR de 10 trames (frames) pour le scenario Best Effort et le scenario DiffServ pour les différentes vidéos compressées avec seulement des trames I, des trames I et P et des trames I, P et B, Ainsi une petite statistique sur les paquets perdus.

 Stefan_cif																	
Nombre de trames						90											
Nombre de trames			852			Nombre de trames			254			Nombre de trames			454		
Compressé avec des trames I.				Compressé avec des trames I et P.				Compressé avec des trames I, P et B.									
Original	Best Effort	DiffServ	Original	Best Effort	DiffServ	original	Best Effort	DiffServ									
38.48	15.30	38.48	38.48	15.30	38.48	38.48	15.30	38.48									
42.26	15.30	42.26	42.28	15.30	42.28	40.07	15.30	17.12									
44.72	15.34	16.73	42.43	15.34	42.43	40.14	15.34	16.04									
42.26	15.34	15.91	42.43	15.34	16.59	42.72	15.34	18.42									
38.54	15.34	15.50	38.37	15.34	15.92	40.27	15.34	16.55									
36.07	15.35	15.29	34.84	15.35	15.60	40.22	15.35	16.30									
33.43	15.34	15.22	32.50	15.34	15.28	42.84	15.34	15.96									
31.57	15.29	15.05	31.03	15.29	15.25	38.36	15.29	15.73									
30.12	15.25	14.79	29.80	15.17	14.99	38.30	15.25	15.52									
28.84	15.28	14.64	28.93	15.08	14.89	42.82	15.28	15.35									

(1)

 Bus_cif																	
Nombre de trames						150											
Nombre de trames			1165			Nombre de trames			396			Nombre de trames			561		
Compressé avec des trames I.				Compressé avec des trames I et P.				Compressé avec des trames I, P et B.									
Original	Best Effort	DiffServ	Original	Best Effort	DiffServ	original	Best Effort	DiffServ									
37.08	10.93	37.08	37.08	10.93	37.08	37.08	10.93	37.08									
41.10	10.90	41.10	41.40	10.90	41.40	38.96	10.90	38.96									
43.95	10.88	17.76	41.52	10.88	41.52	39.01	10.88	17.82									
41.09	10.86	16.46	41.58	10.86	17.58	41.65	10.86	18.01									
37.18	10.85	15.79	37.42	10.85	16.32	39.29	10.85	16.49									
34.73	10.81	15.22	34.22	10.81	15.76	39.31	10.81	15.79									
32.20	10.82	14.87	32.05	10.82	15.12	41.60	10.82	15.14									
30.47	10.83	14.55	30.53	10.83	14.85	39.31	10.83	14.86									
29.21	10.85	14.58	29.41	10.85	14.61	37.60	10.85	14.59									
28.20	10.89	13.97	28.66	10.88	14.53	37.08	10.89	14.13									

(2)

 Foreman_cif								
Nombre de trames						300		
Nombre de trames			Nombre de trames			Nombre de trames		
1533			510			567		
Compressé avec des trames I.			Compressé avec des trames I et P.			Compressé avec des trames I, P et B.		
Original	Best Effort	DiffServ	original	Best Effort	DiffServ	original	Best Effort	DiffServ
39.78	12.06	39.78	39.78	12.06	39.78	39.78	12.06	39.78
44.18	12.08	44.00	42.11	12.08	42.11	40.30	12.08	40.30
44.18	12.10	43.18	42.29	12.10	42.29	40.42	12.10	40.42
44.16	12.13	44.16	42.34	12.13	42.34	42.14	12.13	42.14
38.38	12.15	38.38	40.28	12.15	40.28	40.58	12.15	40.58
36.43	12.15	36.43	37.37	12.15	37.37	40.65	12.15	29.15
34.53	12.30	34.53	36.53	12.30	36.53	42.22	12.15	25.32
33.13	12.31	33.13	35.74	12.31	35.74	40.31	12.14	24.35
32.01	12.34	32.01	35.41	12.34	35.41	39.52	12.14	22.41
31.17	12.42	31.17	34.70	12.42	34.70	42.18	12.13	21.65

(3)


 Hall_cif								
Nombre de trames						300		
Nombre de trames			Nombre de trames			Nombre de trames		
1580			570			592		
Compressé avec des trames I.			Compressé avec des trames I et P.			Compressé avec des trames I, P et B.		
Original	Best Effort	DiffServ	original	Best Effort	DiffServ	original	Best Effort	DiffServ
39.55	12.87	39.55	39.55	12.87	39.55	39.55	12.87	39.55
42.19	12.87	42.19	42.64	12.87	42.64	40.80	12.87	40.80
44.51	12.87	44.51	42.70	12.87	42.70	40.74	12.87	40.74
42.20	12.89	42.20	42.70	12.89	42.70	42.62	12.89	42.62
39.57	12.88	39.57	41.00	12.87	41.00	41.02	12.88	41.02
37.77	12.89	36.94	39.33	12.88	39.33	40.97	12.89	40.97
35.78	12.89	36.43	39.14	12.89	39.14	42.65	12.89	42.65
34.23	10.64	36.14	39.28	12.90	39.28	41.04	12.89	41.04
33.07	10.63	34.51	39.34	12.89	39.34	40.39	12.89	40.39
31.99	10.63	33.40	39.38	12.90	39.38	42.68	12.89	42.68

Figure : Comparaison du rapport PSNR pour 10 premières trames.

Le tableau suivant donne le nombre des trames (frames) perdues pour les différents scénarios (Best effort et DiffServ).

		Best effort	DiffServ
Vidéo	Nombre de trames	Nombre des trames perdues	Nombre des trames perdues
Stefan.yuv	90	75	11
Bus.yuv	150	126	6
Formean.yuv	300		3
Hall.yuv	300	34	2

Tableau : Statistique des paquets perdus pour la compression avec des trames I.

		Best effort	DiffServ
Vidéo	Nombre de trames	Nombre des trames perdues	Nombre des trames perdues
Stefan.yuv	90	15	4
Bus.yuv	150	22	5
Formean.yuv	300	32	0
Hall.yuv	300	36	0

Tableau : Statistique des paquets perdus pour la compression avec des trames I et P.

		Best effort	DiffServ
Vidéo	Nombre de trames	Nombre des trames perdues	Nombre des trames perdues
Stefan.yuv	90	50	26
Bus.yuv	150	59	37
Formean.yuv	300	41	29
Hall.yuv	300	43	28

Tableau : Statistique des paquets perdus pour la compression avec des trames I, P, et B.

Les figures suivantes donnent une visualisation des trames pour les différents scénarios (Best effort, DiffServ)

















	Vidéo Originale	Best Effort	DiffServ
Frame 56			
Frame 57			
Frame 58			
Frame 59			
Frame 60			
Frame 61			

Figure : Comparaison visual entre vidéo STEFAN originale, Best effort et DiffServ (frame 57, 58, 59, 60, 61) (la vidéo est codé avec des trame I.

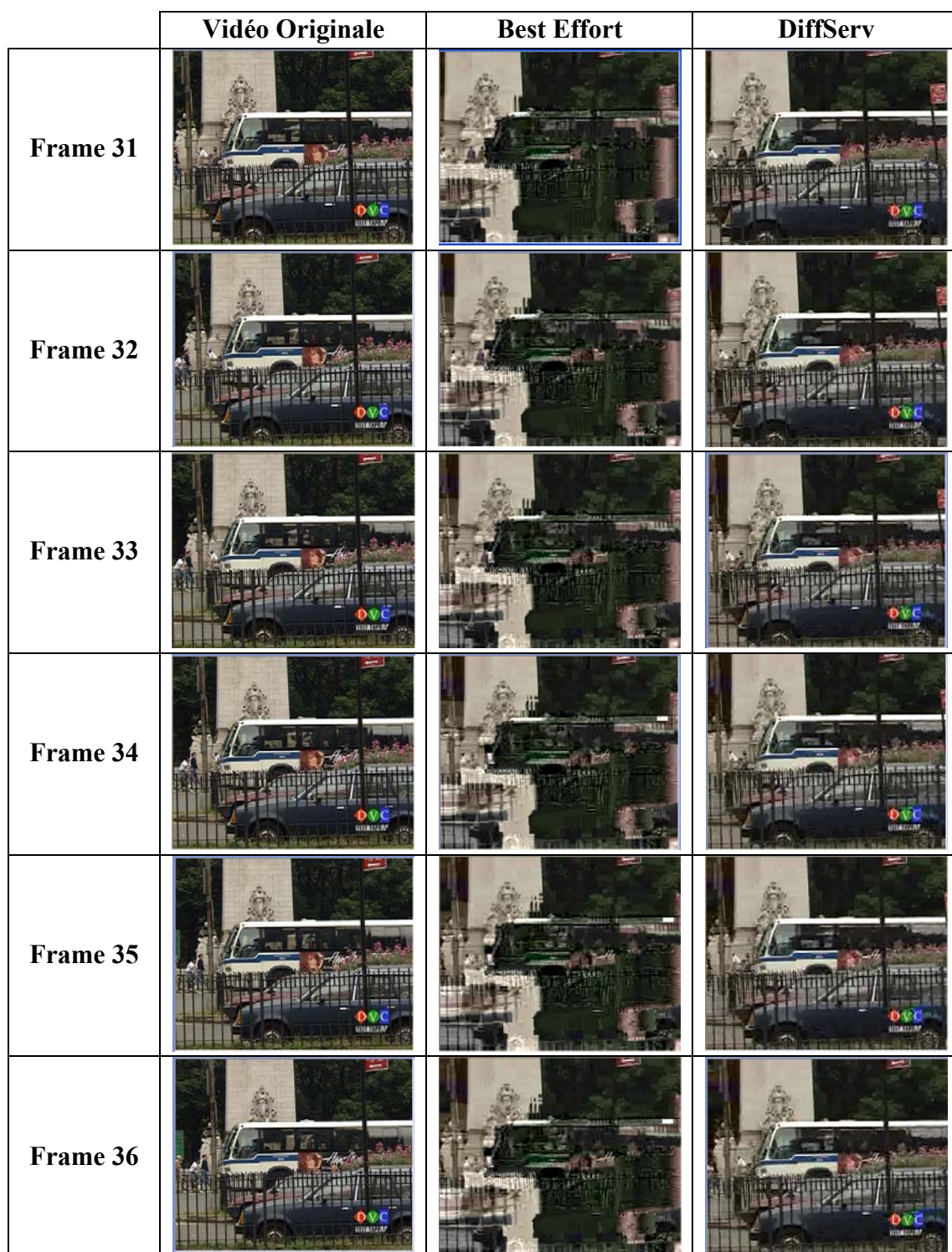


Figure : Comparaison visuelle entre vidéo BUS originale, Best effort et DiffServ (frame 31, 32, 33, 34, 35, 36) (la vidéo est codé avec des trame I, P.


	Originale	Best Effort	DiffServ
Frame 40			
Frame 41			
Frame 42			
Frame 43			
Frame 44			
Frame 45			

Figure : Comparaison visuel entre vidéo FOREMAN originale, Best effort et DiffServ (frame 40, 41, 42, 43, 44, 45)(la vidéo est codé avec des trame I, P et B.

IV.5. Conclusion

Ce travail nous a permis d'acquérir de très intéressantes connaissances concernant : la compression vidéo, la qualité de service et les contraintes qu'elle doit surmonter. Il nous a permis aussi de nous familiariser avec le simulateur réseau NS.

Nous aurions souhaité que les résultats de la simulation soient reportés sur une application avec un réseau réel.

Le thème de ce projet est un sujet d'actualité. Il contient des parties qui font, actuellement, partie du domaine de recherche (Evaluation de la qualité vidéo) et développement (Mécanismes d'ordonnancement et de gestion de files d'attente).

La gestion de la qualité de service pour un opérateur est une exigence pour satisfaire les besoins de l'utilisateur final.

Nous constatons qu'il n'existe pas de modèles parfaits répondant à toutes les exigences d'une bonne qualité de service, donc ce sujet est loin d'être clos.


```

$core shape hexagon
set e1 [$ns node]
$e1 color "blue"
set s2 [$ns node]
$s2 color "black"
#$s2 shape "rectangular"
set s1 [$ns node]
$s1 color "black"
#$s1 shape "rectangular"

# Creation des liens entre les noeud.
$ns duplex-link $e2 $d2 10Mb 20ms DropTail
#affichage de l      fille d'attente
$ns duplex-link-op $e2 $d2 queuePos 0.5
$ns duplex-link-op $e2 $d2 color black
$ns duplex-link-op $e2 $d2 orient 315deg
$ns duplex-link $e2 $d1 10Mb 20ms DropTail
$ns duplex-link-op $e2 $d1 color black
$ns duplex-link-op $e2 $d1 orient 45deg
$ns simplex-link $e2 $core 1Mb 20ms DropTail
$ns simplex-link-op $e2 $core queuePos 0.5
$ns simplex-link-op $e2 $core color green
$ns simplex-link-op $e2 $core orient 180deg
$ns simplex-link $core $e2 1Mb 20ms DropTail
$ns simplex-link-op $core $e2 queuePos 0.5
$ns simplex-link-op $core $e2 color green
$ns simplex-link-op $core $e2 orient 0deg
$ns simplex-link $core $e1 10Mb 20ms DropTail
$ns simplex-link-op $core $e1 queuePos 0.5
$ns simplex-link-op $core $e1 color green
$ns simplex-link-op $core $e1 orient 180deg
$ns simplex-link $e1 $core 10Mb 20ms DropTail
$ns simplex-link-op $e1 $core queuePos 0.5
$ns simplex-link-op $e1 $core color green
$ns simplex-link-op $e1 $core orient 0deg
$ns duplex-link $s2 $e1 10Mb 20ms DropTail
$ns duplex-link-op $s2 $e1 queuePos 0.5
$ns duplex-link-op $s2 $e1 color black
$ns duplex-link-op $s2 $e1 orient 45deg
$ns duplex-link $s1 $e1 10Mb 20ms DropTail
$ns duplex-link-op $s1 $e1 queuePos 0.5
$ns duplex-link-op $s1 $e1 color black
$ns duplex-link-op $s1 $e1 orient 315deg

#configurerles files d'attente ( nombre maximale dans la file d'attente)
set qE1C [[$ns link $e1 $core] queue]
$qE1C set limit_ 10
set qE2C [[$ns link $e2 $core] queue]
$qE2C set limit_ 10
set qCE1 [[$ns link $core $e1] queue]
$qCE1 set limit_ 10
set qCE2 [[$ns link $core $e2] queue]
$qCE2 set limit_ 10
#-----
#la video-----
# source
set udp0 [new Agent/myUDP]
$ns attach-agent $s1 $udp0
$udp0 set packetSize_ $packetSize
$udp0 set_filename emis_be
#Destination-----
set null0 [new Agent/myUdpSink2]
$ns attach-agent $d1 $null0
$ns connect $udp0 $null0
>null0 set_trace_filename recus_be
$udp0 set_fid_ 0

```

```

#-----
#traffic de charge
# Configurer une connexion CBR entre s2 et d2
#Source
set udp1 [new Agent/UDP]
$ns attach-agent $s2 $udp1
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp1
$cbr set packet_size_ $packetSize
$udp1 set packetSize_ $packetSize
$cbr set rate_ $rate
$udp1 set fid_ 1
#-----
# label
$ns at 0.0 "$s1 label Serveur.Vidéo"
$ns at 0.0 "$s2 label CBR"
#$ns at 0.0 "$e1 label Routeur_de_coeur"
$ns at 0.0 "$e2 label Edge-Router"
$ns at 0.0 "$core label Core-Router"
$ns at 0.0 "$d1 label St.s_Réceptrices"
#-----
#$ns add-agent-trace $udp0 Video
#$ns monitor-agent-trace $udp0
#$udp0 tracevar cwnd_
#-----

#Destination-----
set null1 [new Agent/Null]
$ns attach-agent $d2 $null1
$ns connect $udp1 $null1
#-----
set original_file_name ST
set trace_file_name video.dat
set original_file_id [open $original_file_name r]
set trace_file_id [open $trace_file_name w]
set frame_count 0
set last_time 0

#-----Messages d'affichage-----
puts ""
puts "=====
puts "                République Algérienne Démocratique et Populaire"
puts "                Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique"
puts "                Ecole Nationale Polytechnique"
puts "                Département d'electroniques"
puts "                Option Signal & Communication"
puts "                -----"
puts "Mémoire de magistère: "
puts "  Optimisation de la transmissions de MPEG-4 dans un réseau IP par la
methode DiffServ"
puts "=====
puts ""
puts ""
#-----

while {[eof $original_file_id] == 0} {
    gets $original_file_id current_line

    scan $current_line "%d%s%d%s%s%d%s" no_ frametype_ length_ tmp1_ tmp2_ tmp3_
tmp4_ tmp5_
    #affichage du current_line

    #puts "$no_ $frametype_ $length_ $tmp1_ $tmp2_ $tmp3_ $tmp4_ $tmp5_"
    #puts ""
    [expr 1000*1000/25]
}

```

```

set time [expr 1000 * 1000/30]

if { $frametype_ == "I" } { set type_v 1 }
if { $frametype_ == "P" } { set type_v 2 }
if { $frametype_ == "B" } { set type_v 3 }
if { $frametype_ == "H" } { set type_v 1 }

puts $trace_file_id "$time $length_ $type_v $max_fragmented_size"
incr frame_count
}

close $original_file_id
close $trace_file_id
set end_sim_time [expr 1.0 * 1000/30 * ($frame_count + 1) / 1000]
puts "la durée de la Simulation est : $end_sim_time secondes."

set trace_file [new Tracefile]
$trace_file filename $trace_file_name
set video [new Application/Traffic/myTrace2]
$video attach-agent $udp0
$video attach-tracefile $trace_file

$ns at 0.0 "$cbr start"
$ns at 0.0 "$video start"
$ns at $end_sim_time "$video stop"
$ns at $end_sim_time "$cbr stop"
$ns at [expr $end_sim_time + 1.0] "finish"

$ns run

```

Annexe2: Fichier DiffServ.tcl

```

#-----
# Ecole Nationale Polytechniques de El-Harrach Alger
# DiffServ.tcl
# Author: KADIK Rachid.
# Date: Septembre 2008.
#
#
#
#   ----
#   | s1 |-----|
#   ----   10 Mb
#           1 ms
#
#
#
#   ----
#   | s2 |-----|
#   ----   10 Mb
#           1 ms
#
#
#
#   ----
#   | e1 |-----| core |-----| e2 |
#   ----   10 Mb   ----   1 Mb
#           10 ms   ----   10 ms
#
#
#   ----
#   | d1 |-----|
#   ----   10 Mb
#           1 ms
#
#
#   ----
#   | d2 |-----|
#   ----   10 Mb
#           1 ms
#-----

set ns [new Simulator]

#le débit du trafic de charge
#set rate 740000
set rate 840000
#840 kbit/s
#2500000
#taille des paquets du trafic de charge
set packetSize_CBR 1000

set max_fragmented_size 1000
set packetSize 1028
set nf [open DiffServ.nam w]
$ns namtrace-all $nf

$ns color 0 black
$ns color 1 red

proc finish {} {
    puts ""
    puts "-----"
    puts "Fin de la Simulation"
    puts "Email: rachid_kadik@hotmail.com"
    puts ""
    puts "Exucution du nam..."
global ns nf
    exec nam DiffServ.nam &
    exit 0
}

#set nd [open out2.tr w]
#$ns trace-all $nd

#ajouter l'entet UDP (8 Octets) et l'entet IP (20 Octets)
set packetSize1028
#declaration des noeuds-----
# Creation des noeud (stations emetrices, receptrice et des routeurs)
set d2 [$ns node]

```

```

$d2 color "black"
#$d2 shape "rectangular"

set d1 [$ns node]
$d1 color "black"
#$d1 shape "rectangular"

set e2 [$ns node]
$e2 color "blue"

set core [$ns node]
$core color "red"
$core shape hexagon

set e1 [$ns node]
$e1 color "blue"

set s2 [$ns node]
$s2 color "black"
#$s2 shape "rectangular"

set s1 [$ns node]
$s1 color "black"
#$s1 shape "rectangular"

#-----
# label
$ns at 0.0 "$s1 label Serveur.Vidéo"
$ns at 0.0 "$s2 label CBR"
#$ns at 0.0 "$e1 label Routeur_de_coeur"
$ns at 0.0 "$e2 label Edge-Router"
$ns at 0.0 "$core label Core-Router"
$ns at 0.0 "$d1 label St.s_Réceptrices"
#-----

#configuration des liaisons-----
$ns duplex-link $s1 $e1 10Mb 20ms DropTail
$ns duplex-link-op $s1 $e1 queuePos 0.5
$ns duplex-link-op $s1 $e1 color black
$ns duplex-link-op $s1 $e1 orient 315deg

$ns duplex-link $s2 $e1 10Mb 20ms DropTail
#$ns duplex-link-op $s2 $e1 queuePos 0.5
$ns duplex-link-op $s2 $e1 color black
$ns duplex-link-op $s2 $e1 orient 45deg

$ns simplex-link $e1 $core 10Mb 20ms DropTail
$ns simplex-link-op $e1 $core queuePos 0.5
$ns simplex-link-op $e1 $core color green
$ns simplex-link-op $e1 $core orient 0deg

$ns simplex-link $core $e1 10Mb 20ms DropTail
$ns simplex-link-op $core $e1 queuePos 0.5
$ns simplex-link-op $core $e1 color green
$ns simplex-link-op $core $e1 orient 180deg

#$ns simplex-link $core $e2 1Mb 1ms dsRED/core

$ns simplex-link $core $e2 1Mb 20ms dsRED/core
$ns simplex-link-op $core $e2 queuePos 0.5
$ns simplex-link-op $core $e2 color green
$ns simplex-link-op $core $e2 orient 0deg

$ns simplex-link $e2 $core 1Mb 20ms DropTail
$ns simplex-link-op $e2 $core queuePos 0.5

```

```

$ns simplex-link-op $e2 $core color green
$ns simplex-link-op $e2 $core orient 180deg

$ns duplex-link $e2 $d1 10Mb 20ms DropTail
$ns duplex-link-op $e2 $d1 color black
$ns duplex-link-op $e2 $d1 orient 45deg

$ns duplex-link $e2 $d2 10Mb 20ms DropTail
#affichage de l      fille d'attente
$ns duplex-link-op $e2 $d2 queuePos 0.5
$ns duplex-link-op $e2 $d2 color black
$ns duplex-link-op $e2 $d2 orient 315deg

#-----Messages d'affichage-----
puts ""
puts
"=====
puts "          République Algérienne Démocratique et Populaire"
puts "      Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique"
puts "          Ecole Nationale Polytechnique"
puts "          Département d'electroniques"
puts "          Option Signal & Communication"
puts "  -----"
puts "Mémoire de magistère: "
puts"=====
puts "  Optimisation de la transmissions d'un flux Multimédia MPEG-4 dans un
réseau IP
          par la methode DiffServ"
puts"=====
puts "Préparé par:
          KADIK Rachid
          Ingénieur en Télécommunications de l'INT-TIC d'Oran."
puts "  -----"
puts ""
#-----

#configurerles files d'attente

set qCE2 [[ $ns link $core $e2 ] queue]
$qCE2 set limit_ 80
# Set DS RED parameters from Core to Edge2:
$qCE2 meanPktSize $packetSize
$qCE2 set numQueues_ 1
$qCE2 setNumPrec 3
$qCE2 addPHBEntry 10 0 0
$qCE2 addPHBEntry 11 0 1
$qCE2 addPHBEntry 12 0 2
$qCE2 addPHBEntry 0 0 2
$qCE2 configQ 0 0 20 40 0.02
$qCE2 configQ 0 1 10 20 0.10
$qCE2 configQ 0 2 5 10 0.20
#la video-----
# source
set udp0 [new Agent/myUDP]
$ns attach-agent $s1 $udp0
$udp0 set packetSize_ $packetSize
$udp0 set_filename emis_diffserv
#Destination-----
set null0 [new Agent/myUdpSink2]
$ns attach-agent $d1 $null0
$ns connect $udp0 $null0
$null0 set_trace_filename recus_diffserv
$udp0 set fid_ 0
#-----

```



```

#traffic de charge
# Configurer une connexion CBR entre s2 et d2
#Source
set udp1 [new Agent/UDP]
$ns attach-agent $s2 $udp1
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp1
$cbr set packet_size_ $packetSize
$udp1 set packetSize_ $packetSize
$cbr set rate_ $rate
#Destination-----
set null1 [new Agent/Null]
$ns attach-agent $d2 $null1
$ns connect $udp1 $null1
$udp1 set fid_ 1
#-----
set original_file_name ST
set trace_file_name video3.dat
set original_file_id [open $original_file_name r]
set trace_file_id [open $trace_file_name w]

#Marquage semantique des paquets
set frame_count 0
set last_time 0

while {[eof $original_file_id] == 0} {
    gets $original_file_id current_line

    scan $current_line "%d%s%d%s%s%s%d%s" no_ frametype_ length_ tmp1_ tmp2_ tmp3_
tmp4_ tmp5_
    #puts "$no_ $frametype_ $length_ $tmp1_ $tmp2_ $tmp3_ $tmp4_ $tmp5_"

    # 30 frames/sec. if one want to generate 25 frames/sec, one can use set time
[expr 1000*1000/25]
    set time [expr 1000 * 1000/30]

    if { $frametype_ == "I" } { set type_v 1 }
    if { $frametype_ == "P" } { set type_v 2 }
    if { $frametype_ == "B" } { set type_v 3 }
    if { $frametype_ == "H" } { set type_v 1 }

    puts $trace_file_id "$time $length_ $type_v $max_fragmented_size"
    incr frame_count
}

close $original_file_id
close $trace_file_id
set end_sim_time [expr 1.0 * 1000/30 * ($frame_count + 1) / 1000]
puts "la durée de la Simulation est : $end_sim_time secondes."

set trace_file [new Tracefile]
$trace_file filename $trace_file_name
set video [new Application/Traffic/myTrace2]
$video attach-agent $udp0
$video attach-tracefile $trace_file

$ns at 0.0 "$cbr start"
$ns at 0.0 "$video start"
$ns at $end_sim_time "$video stop"
$ns at $end_sim_time "$cbr stop"
$ns at [expr $end_sim_time + 1.0] "$qCE2 printStats"
$ns at [expr $end_sim_time + 1.0] "finish"

$ns run

```

Bibliographie

- [1] A. C. C. Aguiar, C. Hoene, J. Klaue, H. Karl, A. Wolisz, and H. Miesmer. Channel-aware schedulers for voip and mpeg-4 based on channel prediction. to be published at MoMuC, 2003.
- [2] Johan Berts and Anders Persson. Objective and subjective quality assessment of compressed digital video sequences. Master's thesis, Chalmers University of Technology, 1998.
- [3] ITU-R Recommendation BT.500-10. Methodology for the subjective assessment of the quality of television pictures, March 2000.
- [4] Sarnoff Corporation. Jndmetrix-iq software and jnd: A human vision system model for objective picture quality measurements, 2002.
- [5] Project P905-PF EURESCOM. Aquavit - assessment of quality for audio-visual signals over internet and umts, 2000.
- [6] Lajos Hanzo, Peter J. Cherriman, and Juergen Streit. *Wireless Video Communications*. Digital & Mobile Communications. IEEE Press, 445 Hoes Lane, Piscataway, 2001.
- [7] Daniel Hertrich. Mpeg4 video transmission in wireless lans — basic qos support on the data link layer of 802.11b. Minor Thesis, 2002.
- [8] H.Sanneck, W.Mohr, L.Le, C.Hoene, and A.Wolisz. Quality of service support for voice over ip over wireless. *Wireless IP and Building the Mobile Internet*, December 2002.
- [9] ISO-IEC/JTC1/SC29/WG11. Evaluation methods and procedures for july mpeg-4 tests, 1996.
- [10] ISO-IEC/JTC1/SC29/WG11. *ISO/IEC 14496: Information technology - Coding of audiovisual objects*, 2001.
- [11] J. Klaue. Evalvid —<http://www.tkn.tu-berlin.de/research/evalvid/fw.html>.
- [12] J. Klaue, J. Gross, H. Karl, and A. Wolisz. Semantic-aware link layer scheduling of mpeg- 4 video streams in wireless systems. In *Proc. of Applications and Services in Wireless Networks (AWSN)*, Bern, Switzerland, July 2003.
- [13] Weiping Li. Overview of fine granularity scalability in mpeg-4 video standard. *IEEE transaction on circuits and systems for video technology*, March 2001.
- [14] Jens-Rainer Ohm. Bildsignalverarbeitung fuer multimedia-systeme. Skript, 1999.
- [15] ITU-T Recommendations P.910 P.920 P.930. Subjective video quality assessment methods for multimedia applications, interactive test methods for audiovisual communications, principles of a reference impairment system for video, 1996.
- [16] Martyn J. Riley and Iain E. G. Richardson. *Digital Video Communications*. Artech House, 685 Canton Street, Norwood, 1997.
- [17] Cormac J. Sreeman, Jyh-Cheng Chen, Prathima Agrawal, and B. Narendran. Delay reduction techniques for playout buffering. *IEEE Transactions on Multimedia*, 2(2):100–112, June 2000.
- [18] ANSI T1.801.01/02-1996. Digital transport of video teleconferencing / video telephony signals. ANSI, 1996.
- [19] ANSI T1.801.03-1996. Digital transport of one-way video signals - parameters for objective performance assessment. ANSI, 1996.

[20] Stephen Wolf and Margaret Pinson. Video quality measurement techniques. Technical Report 02-392, U.S. Department of Commerce, NTIA, June 2002.

[21] D. Wu, Y. T. Hou, W. Zhu, H.-J. Lee, T. Chiang, Y.-Q. Zhang, and H. J. Chao. On end to end architecture for transporting mpeg-4 video over the internet. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(6):923–941, September 2000

[rfc791] *Internet Protocol Specification*; Information Sciences Institute, USC; septembre 1981.

[rfc1663] *Integrated Services in the Internet Architecture: an Overview*; R. Braden, D. Clark, S. Shenker; Informational RFC; juin 1994.

[rfc1752] *The Recommendation for the IP Next Generation Protocol*; S. Bradner, A. Mankin; Standards Track RFC; janvier 1995.

[rfc1889] *RTP: A Transport Protocol for Real-Time Applications*; H. Schulzrinne, GMD Focus et al; Standards Track RFC; janvier 1996.

[rfc2205] *Resource ReSerVation Protocol (RSVP) - Version 1: Technical Specification*; R. Braden, L. Zhang, et al.; Standards Track RFC; septembre 1997.

[rfc2207] *RSVP Extensions for IPSEC Data Flows*; L. Berger, T. O'Malley; Standards Track RFC; septembre 1997.

[rfc2210] *The Use of RSVP with IETF Integrated Services*; J. Wroclawski; Standards Track RFC; septembre 1997.

[rfc2211] *Specification of the Controlled-Load Network Element Service*; J. Wroclawski; Standards Track RFC; septembre 1997.

[rfc2212] *Specification of Guaranteed Quality of Service*; S. Shenker, C. Partridge, R. Guerin; Standards Track RFC; septembre 1997.

[rfc2215] *General Characterization Parameters for Integrated Service Network Elements*; S. Shenker, J. Wroclawski; Standards Track RFC; septembre 1997.

[rfc2382] *A Framework for Integrated Services and RSVP over ATM*; E. Crawley, L. Berger et al.; Informational RFC; août 1998.

[rfc2474] *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*; K. Nichols, S. Blake, F. Baker, D. Black; Standards Track rfc; décembre 1998.

[rfc2475] *An Architecture for Differentiated Services*; S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss; Standards Track rfc; décembre 1998.

[rfc2581] *TCP Congestion Control*; M. Allman, V. Paxson, W. Stevens; Standards Track RFC; avril 1999.

[22] FFmpeg homepage. <http://ffmpeg.mplayerhq.hu/>, link last active: May 2007.

[23] ns-allinone 2.30 l'URL <http://wwwmash.CS.Berkeley.EDU/ns/>, Dernière mise-à-jour Mai 2008.