

## TABLE DES MATIERES

NOTATIONS.....	iii
INTRODUCTION.....	1
CHAPITRE 1 : LE SIGNAL DE TELEVISION.....	3
1.2.1. Le signal audio.....	3
1.2.2. Le signal vidéo.....	4
1.2.2.1. <i>Le signal vidéo dans l'émission monochrome</i> .....	4
1.2.2.2. <i>Le signal vidéo dans l'émission couleur</i> .....	5
1.2.3. Les signaux de synchronisation:.....	9
1.2.4. Le signal vidéo composite:.....	9
CHAPITRE 2 : ACQUISITION ET TRAITEMENT DU SIGNAL DE TELEVISION.....	11
2.3.1. L'échantillonnage.....	12
2.3.2. La quantification.....	13
2.3.3. La conversion Analogique Numérique.....	13
2.4.1. Les différents types de compression.....	14
2.4.1.1. <i>La compression sans perte de données ou non destructive</i> .....	14
2.4.1.2. <i>La compression avec perte de données (irréversible) ou destructive</i> .....	14
2.4.2. Quelques technologies de compression.....	14
2.4.2.1. <i>Les technologies de compression d'image</i> .....	14
2.4.2.2. <i>Les technologies de compression du son</i> .....	16
CHAPITRE 3 : LA WEB TV.....	20
3.7.1. Définition.....	25
3.7.2. Les buts du Streaming.....	25
3.7.3. Le principe du Streaming.....	25
3.7.4. Le fonctionnement du Streaming.....	25
3.7.5. Les différents types de Streaming.....	26
3.7.5.1. <i>Le Streaming Unicast</i> .....	27
3.7.5.2. <i>Le Streaming Multicast</i> .....	28
3.7.6. Les solutions en Streaming.....	29
3.7.7. Les points forts et les points faibles du Streaming.....	30
CHAPITRE 4 : LES TECHNOLOGIES D'ACCES.....	31
4.2.1. L'ADSL (Asymetric Digital Subscriber Line).....	32
4.2.1.1. <i>Se connecter à Internet via l'ADSL</i> .....	33
4.2.2. Les autres technologies d'accès.....	35
CHAPITRE 5 : LES PROTOCOLES UTILISES.....	38
5.8.1. Le protocole RTP (Real-time Transport Protocol).....	46
5.8.1.1. <i>L'objet du protocole RTP</i> .....	46
5.8.1.2. <i>Les caractéristiques du protocole RTP</i> .....	46
5.8.1.3. <i>Les fonctions du protocole RTP</i> .....	47
5.8.1.4. <i>L'entête du protocole RTP</i> .....	47
5.8.2. Le protocole RTCP (Real-time Transport Control Protocol).....	48
5.8.2.1. <i>L'objet du protocole RTCP</i> .....	48
5.8.2.2. <i>Les fonctions du protocole RTCP</i> .....	48
5.8.2.3. <i>Les formats des paquets</i> .....	49
5.8.3. Le protocole RTSP (Real Time Streaming Protocol).....	52
5.8.3.1. <i>Généralités sur le protocole RTSP</i> .....	52
5.8.3.2. <i>Les principes du protocole RTSP</i> .....	53
5.8.3.3. <i>Les méthodes du protocole RTSP</i> .....	54
CHAPITRE 6 : SIMULATION DE LA MISE EN ŒUVRE D'UNE WEB TV .....	55
6.1.1. Mise en place du réseau local.....	56
6.1.1.1. <i>Installation matérielle</i> .....	56
6.1.1.2. <i>Installation logicielle</i> .....	56
6.1.1.3. <i>Le câblage</i> .....	57
6.1.2. Installation de la carte TV.....	58

<i>6.1.2.1. Installation matérielle</i> .....	58
<i>6.1.2.2. Installation logicielle</i> .....	58
<b>6.1.3. Mise en place du Serveur Web</b> .....	<b>58</b>
<b>6.2.1. Présentation de l'API JMF</b> .....	<b>59</b>
<b>6.2.2. Une brève description de l'API JMF</b> .....	<b>59</b>
<b>6.2.3. Décomposition de l'API JMF</b> .....	<b>59</b>
<i>6.2.3.1. La Gestion des données</i> .....	60
<i>6.2.3.2. Les Managers</i> .....	60
<i>6.2.3.3. Les objets Players</i> .....	61
<i>6.2.3.4. L'objet Processor</i> .....	61
<b>6.2.4. Le fonctionnement de l'API JMF : l'API RTP</b> .....	<b>62</b>
<i>6.2.4.1. Réception d'un flux RTP</i> .....	62
<i>6.2.4.2. Emission d'un flux RTP</i> .....	62
<b>6.2.5. Développement des Applications Clients et Serveurs</b> .....	<b>63</b>
<i>6.2.5.1. WebTVServer 1.0</i> .....	63
<i>6.2.5.2. WebTVPPlayer 1.0</i> .....	65
<i>6.2.5.3. Le fichier de configuration « configure.jsa »</i> .....	68
<b>CONCLUSION</b> .....	<b>71</b>
<b>ANNEXE 1 : LES STANDARDS MONDIAUX DE TELEVISION</b> .....	<b>72</b>
<b>ANNEXE 2 : LES PORTS</b> .....	<b>73</b>
<b>ANNEXE 3 : LES DIALOGUES RTSP</b> .....	<b>75</b>
<b>ANNEXE 4 : QUELQUES PROFILS SOURCES JAVA</b> .....	<b>77</b>
<b>ANNEXE 5 : QUELQUES DEFINITIONS</b> .....	<b>85</b>
<b>BIBLIOGRAPHIE</b> .....	<b>88</b>

## NOTATIONS

AAC	Advanced Audio Coding
API	Application Programming Interface
ADSL	Asymmetric Data Subscriber Line
ATSC	Advanced Television Systems Committee
AVC	Advanced Video Coding
BAS	Broadband Access Server
BGP	Border Gateway Protocol
BLR	Boucle Locale Radio
CAN	Convertisseur Analogique Numérique
CCIR	Comité consultatif International des Radiocommunications
CCITT	Consultatif Committee International Telephone Telegraph
CD	Compact Disc
VPL	Courants Porteurs en Ligne
FM	Frequency Modulation
DAT	Digital Audio Tape
DSB-SC	Double Side Band Suppressed Carrier
DSLAM	Digital Subscriber Line Access Multiplexer
DVB	Digital Video Broadcast
DVD	Digital Versatile Disc
FAI	Fournisseur d'Accès Internet
FCC	Federal Communications Commission
FTP	File Transfer Protocol
GGP	Gateway-Gateway Protocol
HTTP	HyperText Transfer Protocol
IEC	International Electro-technical
IEEE	Institute of Electrical Electronical Engineers
IETF	Internet Engineering Task Force
IMTC	International Multimedia Teleconferencing Consortium
INTERNET	Interconnected Network
IP	Internet Protocol
ISDB-T	Integrated Service Digital Broadcast Terrestrial
ISO	International Standardization Organization

ISP	Internet Service Provider
ITU	International telecommunication Union
JDK	Java Developement Kit
JMF	Java Media Framework
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
MBONE	Multicast Backbone
MJPEG	Motion JPEG
MP3	MPEG-1 Audio Layer 3
MPEG	Motion Picture Expert Group
MPEG-4 AVC	MPEG-4 Advanced Video Coding
NTSC	National Television System Committee
OSI	Open System Interconnection
PAL	Phase Alternance Line
PCI	Peripheral Component Interconnect
RFC	Request For Comment
RSVP	Resource reSerVation protocol
RTC	Réseau Téléphonique Commuté
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SDP	Session Description Protocol
SECAM	Séquence de Couleur Avec Mémoire
SIP	Session Initiation Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UIT-T	Union Internationale des Télécommunications- standardisation du secteur Télécommunication
VoD	Video on Demand
VoIP	Voice over IP
WAN	Wide Area Network
WECA	Wireless Ethernet Compatibility Alliance
WLL	Wireless Local Loop
WMA	Windows Media Audio

## **INTRODUCTION**

Le moyen classique utilisé pour regarder les émissions télévisées est d'avoir un poste de télévision, de le brancher à une prise électrique et à une antenne de réception. Il y a aussi les émissions diffusées par les chaînes payantes dans lesquelles un décodeur est nécessaire : il s'agit de la télévision par satellite, qui exige une antenne parabolique, et de la télévision câblée.

En 1990, l'idée de diffuser les programmes de radio et de télévision sur Internet est née aux Etats-Unis.

Internet n'avait pourtant pas été conçu pour supporter cet aspect de transmission en temps réel, d'autant plus que la vidéo et l'audio sont volumineux et exigent des débits très importants. Sur Internet, le protocole utilisé est HTTP (HyperText Transfer Protocol). Une brève description de ce protocole est que le client envoie sa requête au serveur, ce dernier présente sa réponse, et la communication est interrompue.

La Télévision sur Internet se fait en temps réel et ne peut pas tolérer de telles interruptions. Il fallait donc adapter Internet pour pouvoir faire transiter, en temps réel, l'audio et la vidéo.

De nouveaux protocoles ont par conséquent été élaborés, et une technique appelée « Streaming » a permis de satisfaire à cette exigence.

Actuellement, la Télévision sur Internet est en plein essor en Europe et aux Etats-Unis et elle est communément connue sous le nom de WEB TV.

Ce mémoire est orienté vers l'étude de ce système de WEB TV.

Pour démontrer nos travaux de recherche, notre travail est divisé en six (6) chapitres :

-Chapitre 1 : LE SIGNAL DE TELEVISION ; c'est un chapitre qui permet au lecteur de découvrir quelques côtés techniques du signal de télévision.

-Chapitre 2 : ACQUISITION ET TRAITEMENT DU SIGNAL DE TELEVISION ; ce chapitre met en évidence les principes de base de la numérisation d'un signal, depuis le traitement de ce dernier jusqu'à l'obtention d'une chaîne d'acquisition numérique, puis les techniques de compression qui lui sont appliquées.

-Chapitre 3 : LA WEB TV ; ce chapitre présente l'architecture type d'un système de WEB TV, ses sous-ensembles et la technique du Streaming.

-Chapitre 4 : LES TECHNOLOGIES D'ACCES ; ce chapitre présente les différents types de réseaux pour pouvoir accéder à la Télévision sur Internet.

-Chapitre 5 : LES PROTOCOLES UTILISES ; ce chapitre décrit en détail les protocoles qui sont utilisés pour une transmission en temps réel.

-Chapitre 6 : SIMULATION DE LA MISE EN ŒUVRE D'UN SYSTEME DE WEB TV: ce chapitre est la concrétisation de nos recherches.

## CHAPITRE 1 : LE SIGNAL DE TELEVISION

### 1.1.Introduction [2] [10] [11]

Depuis son invention, l'émission TV n'a pas cessé d'évoluer, une évolution qui se manifeste par des modifications appliquées aux signaux vidéo et audio. L'évolution la plus remarquée est le passage de l'émission monochrome (Noir et Blanc) vers l'émission couleur.

Dans l'émission monochrome, seul un signal dit de *luminance* caractérise l'image et selon les techniques utilisées dans divers pays pour son émission, on définit le *standard* comme D, B, K, K', L, etc.

A l'arrivée de l'émission couleur, des signaux dits de *chrominance* forment, avec le signal de *luminance*, les éléments nécessaires pour une image. Selon les procédés d'émission des signaux de *chrominance*, on définit le *système de couleur* comme NTSC, PAL et SECAM.

Une émission de télévision est donc caractérisée par deux grands paramètres : le *standard* et le *système de couleur*.

Les signaux dits de *luminance*, de *chrominance*, ainsi que d'autres types de signaux qui composent le signal de télévision seront abordés dans les paragraphes plus loin de ce chapitre.

### 1.2.Le signal analogique de télévision [1] [2] [3] [4] [10] [11]

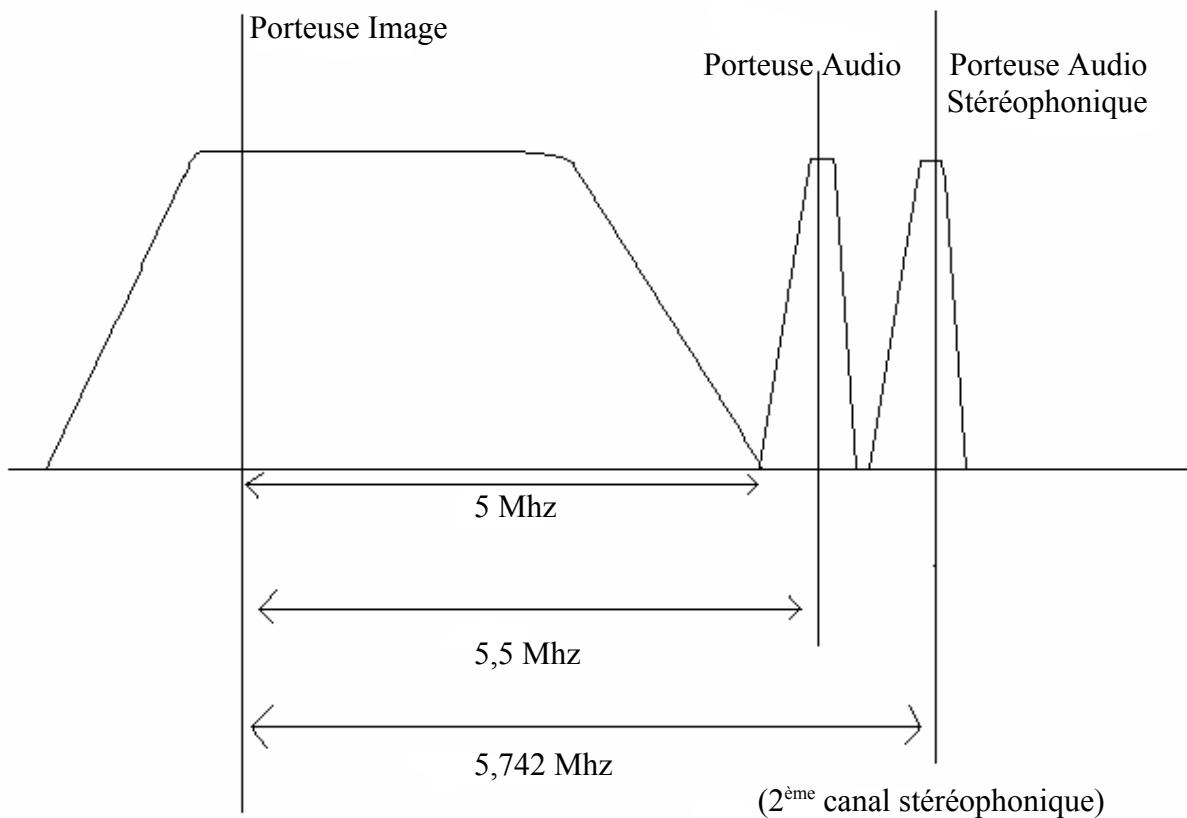
Le signal émis à partir d'un émetteur de TV est divisé en deux parties distinctes :

- la partie audio,
- la partie correspondant aux informations image : le signal de *luminance*, le signal de *chrominance* et les signaux de synchronisation.

#### 1.2.1. *Le signal audio*

Le signal audio occupe la bande de fréquence de 40 Hz à 15 Khz. Il doit être reproduit en phase avec l'image.

En général, le signal audio en TV est unique, mais certains standards comme B et G autorisent l'émission d'un ou plusieurs signaux audio. Ces signaux pourront être le canal gauche et droite pour un son en stéréophonie par exemple, comme le montre la *Figure 1.01* :



*Figure 1.01 : exemple de spectre de fréquence utilisé pour les normes B et G.*

### **1.2.2. Le signal vidéo**

Le signal vidéo diffère de par sa structure selon que l'émission se fait, soit en monochrome, soit en couleur.

#### **1.2.2.1. Le signal vidéo dans l'émission monochrome**

Dans le cas d'une émission monochrome, l'image est en noir et blanc. Elle peut être reproduite à partir de la luminance des différentes parties qui la composent, c'est-à-dire de leur caractère clair ou sombre. A chaque valeur du signal vidéo (désigné par  $Y$  ou  $E_Y$ ) correspond une nuance de gris comprise entre le blanc et le noir. Le signal de luminance est encadré de tops de synchronisation ligne qui précisent la position des bords gauche et droite de l'écran comme indiquée sur la *Figure 1.02* :

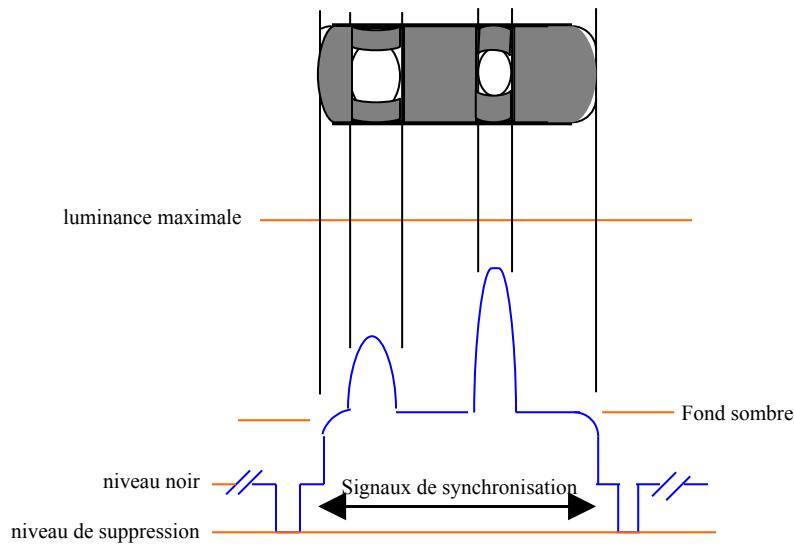


Figure 1.02 : variation de la luminance en fonction des points d'une image.

### 1.2.2.2. Le signal vidéo dans l'émission couleur

Le système couleur repose sur la décomposition d'une image colorée en trois composantes couleurs fondamentales : Rouge-Vert-Bleu.

Comme en système monochrome, on doit assurer la transformation de ces images d'une seule teinte en trois signaux électriques désignés souvent par  $E_R$  pour l'image rouge,  $E_V$  pour l'image verte et  $E_B$  pour l'image bleue. On n'émet pas directement et intégralement ces signaux mais plutôt leur différence entre la luminance ; les signaux  $(E_B-E_Y)$  et  $(E_R-E_Y)$  représentent les bases des signaux de chrominance.

Avec ces signaux de différence, l'amplitude des signaux de chrominance est faible devant l'amplitude du signal de luminance et cet aspect contribue à protéger le récepteur noir et blanc des reflets chatoyants lorsqu'il s'agit d'observer l'image en noir et blanc.

#### 1.2.2.2.1. Le signal de luminance dans l'émission couleur

A l'arrivée du système couleur, la condition de compatibilité est exigée : un récepteur TV noir et blanc doit pouvoir recevoir en noir et blanc, une émission couleur. Un signal est donc nécessaire au récepteur noir et blanc, c'est le signal de luminance en émission couleur. Il est obtenu par la combinaison des signaux  $E_V$ ,  $E_B$ ,  $E_R$ , selon la proportion suivante :

$$E_Y = 0,30 E_R + 0,59 E_V + 0,11 E_B \quad (1.01)$$

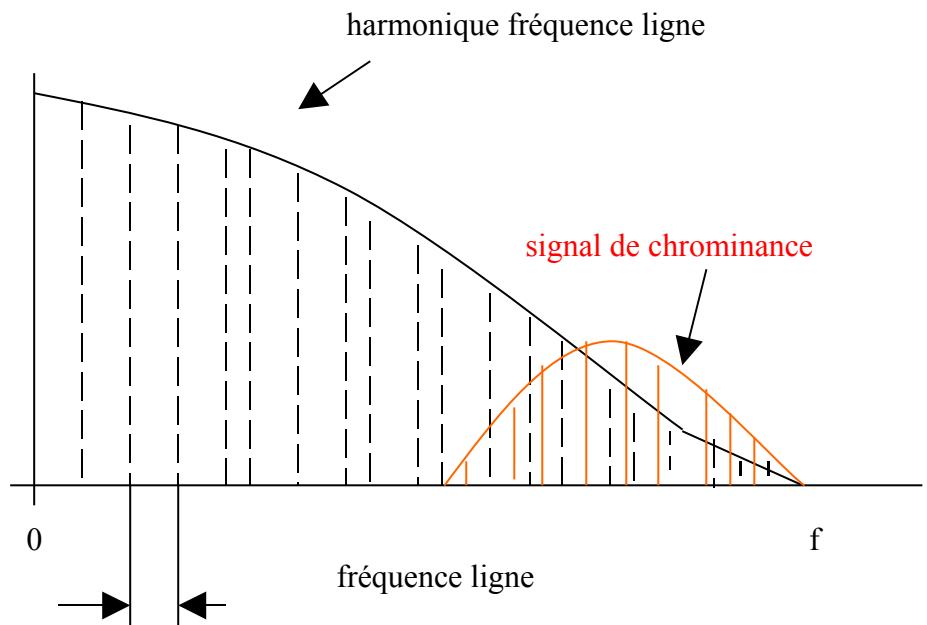
Ces proportions sont valables pour tous les systèmes couleurs. On remarque ici l'avantage des signaux de différence ; En effet, ces différences deviennent nulles chaque fois que le point de l'image explorée est blanc, gris ou noir : les signaux de chrominance disparaissent.

Par exemple, pour un point blanc,  $E_R=E_V=E_B=1$  et  $E_Y=1$ , d'où  $E_B - E_Y=0$  et  $E_R - E_Y=0$ .

#### *1.2.2.2.2. Les signaux de chrominance en émission couleur*

Comme il faut rajouter au signal de luminance les deux signaux de base de couleur ( $E_Y - E_B$ ) et ( $E_Y - E_R$ ) et qu'il n'y a qu'une seule voie, un codage doit être entrepris. C'est là qu'interviennent les systèmes et leur propre originalité. Dans tous les systèmes de télévision couleur, les signaux de chrominance sont insérés dans le haut du spectre de luminance pour diminuer encore l'influence sur la luminance. Et l'information de chrominance est portée par une sous-porteuse chroma modulée par les signaux de couleur.

La *Figure 1.03* montre l'allure du spectre du signal de TV couleur.



*Figure 1.03 : allure du spectre du signal de TV couleur.*

#### *1.2.2.2.3. Les signaux de chrominance dans le système NTSC*

Divers paramètres, tels que la sensibilité de l'œil, le respect du niveau du signal pour éviter la surmodulation, ont été pris en compte pour aboutir aux composantes du signal de chrominance dans le système NTSC :

$$E_I = 0,74 \cdot (E_R - E_Y) - 0,27 \cdot (E_B - E_Y) = 0,21 E_R - 0,52 E_Y + 0,31 E_R \quad (1.02)$$

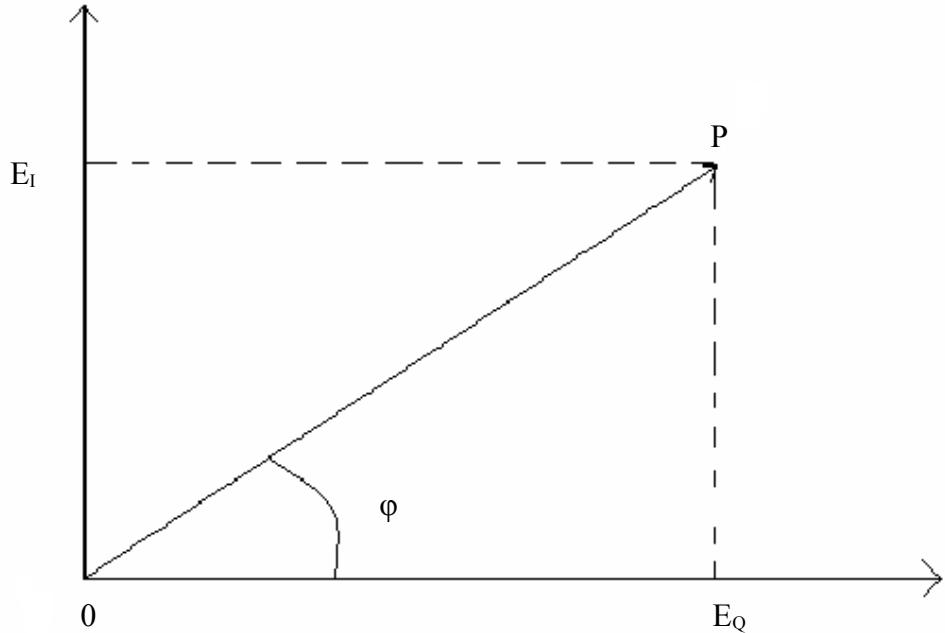
$$E_Q = 0,48 \cdot (E_R - E_Y) - 0,41 \cdot (E_B - E_Y) = 0,6 E_R - 0,28 E_Y - 0,32 E_R \quad (1.03)$$

Une onde sinusoïdale de fréquence  $f_{sc} = 3,58$  MHz est modulée selon DSB-SC (Double Side Band Suppressed Carrier) par le signal  $E_Q$ .

Une autre onde de même fréquence, mais déplacée de  $90^\circ$  par rapport à la précédente, est modulée de la même manière par le signal  $E_I$ . La superposition des deux signaux donne l'information de chrominance. L'expression du signal résultant est alors :

$$E_C = OP(t) = E_I \sin(2\pi f_{sc}t) + E_Q \cos(2\pi f_{sc}t) \quad (1.04)$$

En utilisant la représentation vectorielle de la modulation d'amplitude, on a la *Figure 1.04* suivante :



*Figure 1.04 : chrominance dans le système NTSC.*

*Remarque :* Une variation de phase peut survenir le long de la transmission, ce qui pourrait modifier la teinte de l'image à la réception : c'est l'inconvénient du système NTSC.

#### 1.2.2.2.4. Les signaux de chrominance dans le système PAL

Dans le système PAL, les composantes du signal de chrominance sont les suivantes :

$$E_U = 0,49 \cdot (E_B - E_Y) \quad (1.05)$$

$$E_V = 0,87 \cdot (E_R - E_Y) \quad (1.06)$$

Le signal  $E_U$  module en DSB-SC une onde de fréquence 4,43 MHz, et  $E_V$  (ou  $-E_V$ ) module de la même manière une onde de même fréquence mais déphasée de  $90^\circ$  et la superposition de ces signaux donne l'information de chrominance. L'expression du signal de chrominance est alors :

$$E_C = E_U \sin(2\pi f_{sc}t) \pm E_V \cos(2\pi f_{sc}t) \quad (1.07)$$

Le système PAL est une amélioration du système NTSC en apportant une méthode pour éliminer l'erreur de phase mais l'amplitude du signal de chrominance résultant est beaucoup plus importante qu'en système NTSC, ce qui confère au système PAL, une forte saturation en couleur.

#### 1.2.2.2.5. Les signaux de chrominance dans le système SECAM

Dans le système SECAM, les composantes du signal de chrominance sont désignées par  $D_B$  et  $D_R$  qui sont définies comme suit :

$$D_B = -1,9 \cdot (E_B - E_Y) \quad (1.08)$$

$$D_R = 1,5 \cdot (E_R - E_Y) \quad (1.09)$$

A la différence avec les systèmes NTSC et PAL, le système SECAM utilise une émission séquentielle de signaux de chrominance. En effet, dans ce système, on émet alternativement le signal  $D_B$  et  $D_R$  pour deux lignes consécutives. On dénomme alors une ligne comme « rouge » lorsque  $D_R$  est émis, et « bleue » lorsque c'est  $D_B$  qui est émis.

Dans le cas d'une ligne rouge, le signal  $D_R$  module en fréquence une porteuse de fréquence  $f_{OR}=4,40625$  Mhz avec une excursion de fréquence  $\Delta f_{OR}=280$  Khz et pour une ligne bleue,  $D_B$  module en fréquence une porteuse de fréquence  $f_{OB} = 4,25$  Mhz avec une excursion de fréquence  $\Delta f_{OB} = 230$  Khz.

$$E_C = 0,07 \cos 2\pi [f_{OR} + \Delta f_{OR}(\int D_R dt)] \text{ pour la ligne « rouge »} \quad (1.10)$$

$$EC=0,07 \cos 2\pi [f_{OB} + \Delta f_{OB}(\int D_B dt)] \text{ pour la ligne « bleue »} \quad (1.11)$$

Dans tous les systèmes couleur, sur le palier arrière de suppression ligne, on transmet un signal alternatif sinusoïdal de fréquence  $f_{OR}$  ou  $f_{OB}$ , pour identifier que le signal de chrominance émis pendant cette ligne est  $D_R$  ou  $D_B$ .

#### **1.2.3. Les signaux de synchronisation:**

Pour assurer la synchronisation entre le système d'acquisition (le balayage de l'écran) et le système de reproduction (tube cathodique à l'écran), on ajoute des informations supplémentaires, dites de synchronisations au signal vidéo résultant de l'image. On distingue alors : les signaux de synchronisation lignes, les signaux de synchronisation trames.

#### **1.2.4. Le signal vidéo composite:**

Le signal vidéo composite  $E_M$  représente l'information intégrale de l'image ; il est composé d'un signal de luminance, de chrominance, ainsi que de divers signaux de synchronisation.

Ainsi, pour le :

Système NTSC :

$$E_M = E_Y + E_I \sin (2\pi f_{sc}t) + E_Q \cos (2\pi f_{sc}t) \quad (1.12)$$

avec  $f_{sc}=3,58$  Mhz.

Système PAL :

$$E_M = E_Y + E_U \sin (2\pi f_{sc}t) \pm E_V \cos (2\pi f_{sc}t) \quad (1.13)$$

avec  $f_{sc}=4,43$  Mhz.

Système SECAM :

$$E_M = E_Y + 0,07 \cos 2\pi [f_{OR} + \Delta f_{OR}(\int D_R dt)] \text{ pour la ligne « rouge »} \quad (1.14)$$

avec  $f_{OR}=4,40625$  Mhz et  $\Delta f_{OR} = 280$  KHz.

et

$$E_M = E_Y + 0,07 \cos 2\pi [f_{OB} + \Delta f_{OB}(\int D_B dt)] \text{ pour la ligne « bleue »} \quad (1.15)$$

avec  $f_{OB}=4,25$  Mhz et  $\Delta f_{OB} = 230$  KHz.

### **1.3.Le signal numérique de télévision [15] [16]**

De nombreuses normes d'émission numérique sont aujourd'hui admises dans divers pays : l'ATSC (Advanced Television Systems Committee) aux Etats-Unis et au Canada, l'ISDB-T (Integrated Service Digital Broadcast Terrestrial) au Japon et le DVB (Digital Video Broadcast) en Europe.

Ces normes d'émission sont liées au développement des normes de compression.

Le CCIR définit, dans la courbe CCIR 601, une vidéo compressée avec une résolution d'entrée de 720x576. Lorsque le taux de compression diminue, la qualité vidéo en sortie se détériore graduellement jusqu'à un peu moins de 5 Mbps, taux en-dessous duquel la qualité chute. D'un autre côté, la courbe SIF (Source Input Format) est incapable de fournir une qualité équivalente à de hauts débits, mais lorsque le taux tombe en-dessous d'environ 3,5 Mbps, elle s'avère plus performante que la courbe CCIR 601. SIF est une résolution réduite obtenue en faisant passer l'entrée de 720x576 à 360x288 avant compression.

Par conséquent :

- en dessous de 3,5 Mbps : la norme MPEG-1 est choisie pour une résolution SIF
- au-dessus de 5 Mbps : la norme MPEG-2 est choisie pour une résolution CCIR 601.

MPEG-2 fonctionne donc à des taux plus élevés que MPEG-1 mais avec un taux de transfert maximum de 15 Mbps, et il permet l'affichage de séquences jusqu'à 30 images par seconde en télévision numérique.

## CHAPITRE 2 : ACQUISITION ET TRAITEMENT DU SIGNAL DE TELEVISION

### 2.1 Les récepteurs de télévision

Selon les systèmes, on a les récepteurs noir et blanc, les récepteurs NTSC, les récepteurs PAL, les récepteurs SECAM, et les récepteurs multi systèmes.

### 2.2 La carte TV

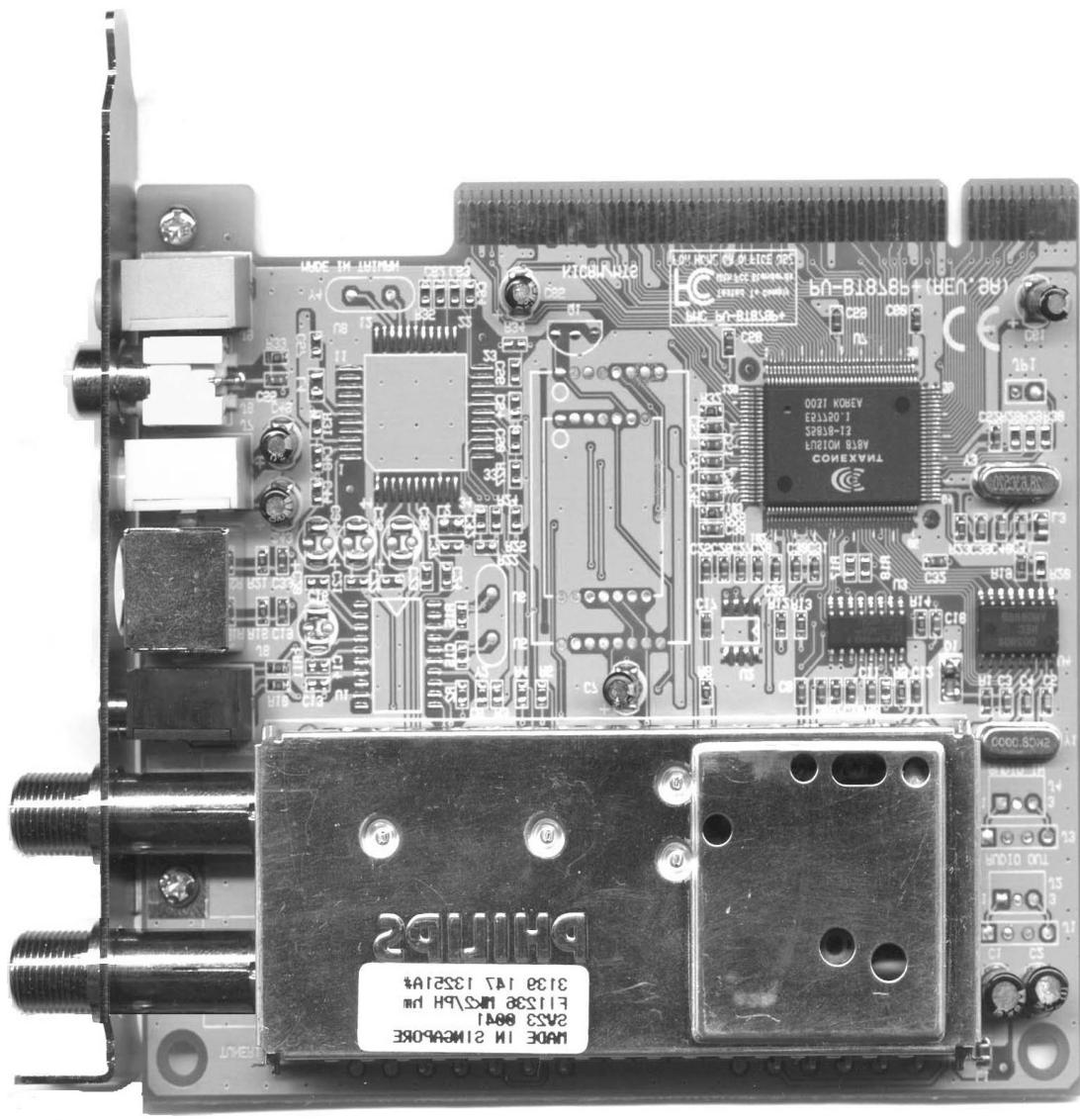


Figure 2.01 : une carte TV.

Une carte TV, voir *Figure 2.01*, est un récepteur multi systèmes. Pour ce mémoire, nous avons utilisé une carte TV PV-BT878P+ w/FM. Celle-ci est composée d'un tuner intégré. Elle permet de capter les émissions TV ainsi que les émissions radiophoniques et comporte une entrée pour antenne CATV/VHF/UHF, une entrée S-Video, une entrée radio FM, une entrée Vidéo composite et une sortie audio.

Elle se branche sur un connecteur PCI (Peripheral Component Interconnect); le signal de télévision est numérisé, puis acheminé vers les bus; C'est ce flux qui est dirigé par les pilotes pour être affiché sur le moniteur d'un PC, ou transmis vers un réseau grâce à des programmes, comme c'est le cas dans la partie simulation de ce mémoire.

Notons qu'il y a aussi un équipement spécifique qui peut être utilisée, la carte de réception à la norme DVB, qui peut être intégrée au micro-ordinateur et reliée à une antenne parabolique extérieure.

### **2.3 La numérisation d'un signal [12]**

Soit  $m(t)$  un signal analogique,  $f_e$  la fréquence d'échantillonnage ;  $m(t)$  est quantifié suivant une suite discrète de niveaux  $m_0, m_1, \dots, m_n$ , tel que  $m_1 - m_0 = q = S_0$ .

#### **2.3.1. L'échantillonnage**

Un signal échantillonné  $m^*(t)$  est obtenu en prélevant sur  $m(t)$ , une suite d'échantillons tel que l'on a :

$$m^*(t) = \{m(0), m(T_e), m(2T_e), \dots\} = m(t) \cdot \sum \delta_n(t - nT_e). \quad (2.01)$$

D'après le Théorème de Shannon, un filtrage passe-bas permet de reconstituer le signal  $m(t)$  à partir de  $m^*(t)$  à condition que l'on ait :

$$f_e \geq 2 f_{max} \quad (2.02)$$

où  $f_{max}$  apparaît dans la décomposition maximale de  $f_e$ .

L'oreille humaine perçoit les fréquences jusqu'à 20 Khz, la fréquence d'échantillonnage doit donc être au moins de l'ordre de 40 Khz. Si l'on prend une fréquence d'échantillonnage qui n'est pas suffisamment élevée par rapport à la fréquence du signal original, on obtient ce que l'on appelle un repli spectral. Cet effet est assimilable à un bruit de fond ou à de la distorsion.

Voici quelques exemples de fréquence d'échantillonnage normalisée :

- 32 Khz : pour la radio FM en numérique
- 44,1 Khz : pour l'audio professionnel et le CD audio
- 48 Khz : pour certains enregistreurs numériques professionnels et l'enregistrement grand public (DAT, Digital Audio Tape).
- Jusqu'à 192 Khz pour le DVD.

### 2.3.2. La quantification

La plage de variation du signal  $m(t)$  est découpée en  $M$  niveaux  $m_0, m_1, \dots, m_{M-1}$  avec  $M=2^N$ , et  $N$  est un nombre entier. Chaque valeur d'échantillon de  $m(t)$  est alors approximée par le niveau dont elle est le plus proche. Le passage du signal échantillonné  $m^*(t)$  au signal quantifié introduit une dégradation du signal analogue à celle que produirait un bruit superposé à  $m^*(t)$ . C'est le bruit de quantification, noté  $\eta(t)$ .

### 2.3.3. La conversion Analogique Numérique

La conversion Analogique Numérique se fait selon le processus ci-dessous.

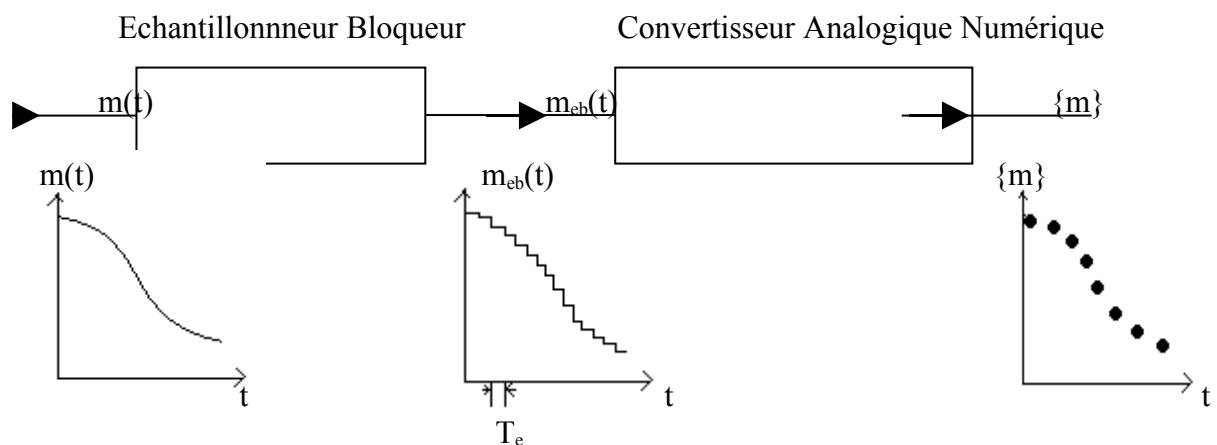


Figure 2.02 : la conversion analogique numérique.

Dans la Figure 2.02 ci-dessus,  $m_{eb}(t)$  est le signal après avoir traversé l'échantillonneur bloqueur.

La carte TV vue précédemment opère ainsi pour fournir le signal numérique et l'achemine vers les bus.

## **2.4 La compression [21] [22] [23] [24]**

La compression a pour objectif de réduire le flot d'informations transmis par unité de temps, tout en préservant le contenu informationnel du signal. Elle élimine les redondances dans le signal, et selon les cas, utilise des systèmes de codage plus performants et mieux adaptés.

Nous n'évoquerons ici que les principes généraux de la compression sans entrer dans les détails techniques.

### ***2.4.1. Les différents types de compression***

On distingue deux types de compression :

- La compression sans perte de données,
- La compression avec perte de données.

#### **2.4.1.1. La compression sans perte de données ou non destructive**

Elle correspond à un ensemble d'algorithmes préservant les données à travers le processus de compression et de décompression. Les codecs (cf ANNEXE 5) non destructifs sont chers et difficiles à développer, ils requièrent beaucoup de puissance machine et le résultat en terme de gain de place n'est pas très bon.

#### **2.4.1.2. La compression avec perte de données (irréversible) ou destructive**

Elle élimine des bits, parfois sans perte de qualité. La compression destructive repose sur le fait que l'être humain ne perçoit pas les fréquences au-delà de 20 KHz et aussi sur le fait que certaines fréquences sont masquées par d'autre. Un autre fait est que l'œil humain n'est pas sensible à certaines dégradations, et ne peut faire une distinction parfaite entre les 16777216 couleurs.

### ***2.4.2. Quelques technologies de compression***

Chaque outil de compression applique ses propres méthodes, c'est pourquoi on trouve aujourd'hui de nombreux formats de compression ; on va en citer quelques-uns :

#### **2.4.2.1. Les technologies de compression d'image**

##### ***2.4.2.1.1. La compression JPEG***

JPEG désigne un groupe, le Joint Photographic Experts Group, qui a mis au point des algorithmes de compressions d'images, avec ou sans perte d'information. Un des principes de la compression JPEG est de prendre en compte toutes les répétitions (des pixels proches ont souvent les mêmes caractéristiques de couleur) ou de mémoriser uniquement les différences d'un pixel à l'autre. De la sorte, l'image peut être reconstituée.

Historiquement, JPEG est une norme qui a été développée par les organismes de normalisations CCITT (Consultatif Committee International Telephone Telegraph), devenu l'ITU (International Telecommunication Union) et l'ISO (International Standardization Organization).

En effet, en 1982, l'ISO a formé le PEG (Photographic Experts Group) dans le but de rechercher des méthodes de transmissions de vidéo, d'images et de texte à travers RNIS.

En 1986, un sous-groupe du CCITT a commencé à rechercher des méthodes de compression de couleur et de niveaux de gris pour des transmissions de fax. Les méthodes de compression utilisées étaient très similaires de celles recherchées par le PEG.

En 1987, le CCITT et l'ISO combinèrent leurs deux groupes dans un comité commun pour développer un standard de compression d'image. Ce nouveau comité avait le nom de JPEG.

JPEG a fourni un format capable de stocker des images en millions de couleurs tout en conservant une très bonne efficacité en matière de compression.

#### *2.4.2.1.2. La compression MJPEG*

MJPEG (Motion JPEG) est une norme conçue pour comprimer des images animées. L'idée de base était d'afficher rapidement une image superposée sur une autre pour donner l'apparence que l'objet à l'intérieur de l'image est en train de bouger.

Le principe du Motion JPEG (noté MJPEG ou M-JPEG) consiste à appliquer successivement l'algorithme de compression JPEG aux différentes images d'une séquence vidéo.

Etant donné que le M-JPEG code séparément chaque image de la séquence il permet d'accéder aléatoirement à n'importe quelle partie d'une vidéo. Ainsi, son débit de 8 à 10 Mbps le rend utilisable dans les studios de montage numérique.

En terme de compression d'image en vue de transmission vidéo en temps réel, la compression JPEG est la plus utilisée.

#### 2.4.2.2. Les technologies de compression du son

Plusieurs normes de compression existent, à savoir la famille G 7x (G 711, G 722, G 728, G 729), le WMA (Windows media Audio) de Microsoft et IBM, le format QuickTime de Apple, le format RealAudio de RealNetworks, la famille MPEG.

La famille MPEG est la plus adoptée en terme de compression.

Nous accorderons une place spécifique à sa description.

MPEG (Moving Picture Expert Group) est une association qui définit une norme de codage et de compression des images, vidéos et sons. Les fichiers MPEG ont l'extension mpg, mp2, mp3, mp4 ou mp7.

Le groupe MPEG a travaillé sous la direction conjointe de l'ISO et de l'IEC (International Electro-technical Commission) afin de mettre en place des standards de compression pour l'audio et la vidéo. Le nom officiel de la norme MPEG est : ISO/IEC JTCI SC29 WG11.

Ce format est spécifié par 3 couches, appelées *layer* ou encore *niveau1*, *niveau2*, *niveau3*. Du niveau1 au niveau3, les algorithmes d'encodage sont de plus en plus complexes.

Il existe plusieurs standards MPEG :

- le **MPEG-1**, développé en 1988, est un standard pour la compression des données vidéo et des canaux audio associés (jusqu'à 2 canaux pour une écoute stéréo). Il permet le stockage de vidéos à un débit de 1.5Mbps dans une qualité proche de celle des cassettes VHS sur un support CD appelé VCD (*Vidéo CD*).

La norme MPEG-1 représente chaque image comme un ensemble de blocs 16 x 16. Elle permet d'obtenir une résolution de 352x240 à 30 images par seconde en NTSC et 352x288 à 25 images par seconde en PAL/SECAM.

Le MPEG-1 permet d'obtenir des débits de l'ordre de 1.2 Mbps. Il permet d'encoder une vidéo grâce à plusieurs techniques :

-Intra coded frames: les images sont codées séparément sans faire référence aux images précédentes.

-Predictive coded frames: les images sont décrites par différence avec les images précédentes.

-Bidirectionally predictive coded frames: les images sont décrites par différence avec l'image précédente et l'image suivante.

-DC Coded frames: les images sont décodées en faisant des moyennes par bloc.

- le **MPEG-2**, un standard dédié originellement à la télévision numérique offrant une qualité élevée à un débit pouvant aller jusqu'à 40 Mbps, et 5 canaux audio. Le MPEG-2 permet de plus une identification et une protection contre le piratage. Il s'agit du format utilisé par les DVD vidéo.
- le **MPEG-4**, un standard destiné à permettre le codage de données multimédia sous formes d'objets numériques, afin d'obtenir une plus grande interactivité, ce qui rend son usage particulièrement adapté au Web et aux périphériques mobiles.
- le **MPEG-7**, un standard visant à fournir une représentation standard des données audio et visuelles afin de rendre possible la recherche d'information dans de tels flux de données. Ce standard est ainsi également intitulé *Multimedia Content Description Interface*.
- le **MPEG-21**, en cours d'élaboration, dont le but est de fournir un cadre de travail (en anglais *framework*) pour l'ensemble des acteurs du numérique (producteurs, consommateurs, ...) afin de standardiser la gestion de ces contenus, les droits d'accès, les droits d'auteurs, ...

#### 2.4.2.2.1. Le MPEG-1

Sa description a été vue plus haut.

#### 2.4.2.2.2. Le MP3 (MPEG-1 Audio Layer 3)

MP3 est un format de compression de données développé par l'ISO: une minute de musique de qualité CD représente 10 Mo. Après une compression MP3, elle en occupe douze fois moins, sans vraiment sacrifier la qualité audio; le taux de compression est dit de 1 : 12. Les fichiers ainsi comprimés portent l'extension mp3.

La compression MP3 exploite la destruction de données n'altérant que faiblement le son pour l'oreille humaine.

Elle consiste à retirer, des données audio, les fréquences inaudibles pour l'auditeur moyen dans des conditions habituelles d'écoute.

L'oreille humaine est capable de discerner, en moyenne, des sons entre 20 Hz et 20 Khz, sachant que sa sensibilité est maximale pour des fréquences entre 2 et 5 Khz (la voix humaine est entre 500 Hz et 2 Khz).

Le standard MP3 reste le standard le plus médiatisé en terme de compression audio.

#### *2.4.2.2.3. Le MP3Pro*

C'est un format développé par Thomson Multimedia et Fraunhofer Institute. Il comprime deux fois plus que le MP3 classique tout en assurant, en théorie, la même qualité sonore.

#### *2.4.2.2.4. Le MP4*

Le format AAC (Advanced Audio Coding) est un codec audio basé sur la norme Mpeg4, et on l'a surnommé MP4.

Ce nouveau standard serait vu comme le successeur du célèbre MP3.

Des experts en matière d'écoute de chez Apple estiment qu'il est quasiment impossible de distinguer à l'oreille les fichiers audio AAC compressés à 128 kbps (stéréo) d'une source audio non compressée.

Le format AAC offre une qualité d'écoute bien supérieur au MP3, plus proche de l'original.

Il est Gapless, c'est-à-dire qu'il n'y aura pas de petite coupure entre 2 morceaux (contrairement au Mp3).

MPEG-4 s'adresse à la fusion de trois mondes : l'informatique, la télécommunication et la télévision.

#### *2.4.2.2.5. Le MP7 (Multimedia Content Description Interface)*

C'est la version 7 de la norme de compression MPEG, sortie officiellement en 2000. Ce nouveau membre de la famille MPEG, appelé *Multimedia Content Description Interface* (MPEG-7) étendra les capacités de recherche limitées d'aujourd'hui pour inclure d'autres types d'information. En d'autres termes, MPEG-7 va spécifier une description standard de différents types d'informations multimédia. Cette description devra être associée au contenu lui-même pour permettre la recherche rapide et efficace des informations qui intéressent l'utilisateur.

## **2.5 L'encodage du signal vidéo**

L'encodage c'est la conversion d'un format en un autre.

La qualité de l'image dépend fortement du soin apporté à l'encodage vidéo. Les fournisseurs de contenu livrent leurs programmes en direct aux prestataires techniques. Ces derniers encodent en direct ces flux audio et vidéo. C'est la partie délicate de l'opération, celle qui peut générer des problèmes de netteté, de pixellisation désagréable lorsque la compression est excessive.

## CHAPITRE 3 : LA WEB TV

### 3.1 La problématique [6]

La diffusion de programmes de radio et de télévision sur Internet est l'un des enjeux majeurs de ce début de siècle. Le concept est né aux USA dans la première moitié des années 90.

A l'origine, Internet n'a pas été conçu pour la vidéo. Ses caractéristiques pourraient même paraître inconciliables avec les exigences de ce média : les accès et les réseaux sont divers et hétérogènes, les débits y sont essentiellement variables et pas toujours suffisamment élevés, alors que la vidéo exige par nature un flux important, continu et régulier. Des progrès significatifs ont été atteints ces dernières années. Des technologies spécifiques ont été développées et aujourd'hui le marché de la télévision sur Internet ou WEB TV est en pleine explosion même si de nombreux progrès restent encore à faire pour améliorer la qualité des images. Les progrès technologiques réalisés au niveau du traitement des données (les micro-ordinateurs sont plus rapides et plus puissants, les algorithmes d'encodage et les logiciels de lecture sont plus performants), le développement à plus grande échelle des accès à hauts débits... sont les éléments déterminants de cette évolution.

### 3.2 L'architecture d'une WEB TV [5]

Une architecture type d'une WEB TV est représentée sur la *Figure 3.01*.

Pour la diffusion des programmes de TV sur Internet, on procède dans le centre d'ADMINISTRATION et de CREATION aux étapes de production pour l'acquisition du son et image, de postproduction pour le traitement de ces produits et finalement à l'étape de diffusion proprement dite sur Internet.

Le centre d'ADMINISTRATION/CREATION est destiné à la préparation des programmes, et ainsi que leur archivage ou BACKUP, toujours recommandé.

Le centre de diffusion sur Internet comprend les serveurs WEB et serveurs STREAMING. Il est en général installé dans une agglomération où l'on peut jouir du haut débit.

De son côté, le client peut s'y connecter de deux façons :

- soit en utilisant un browser comme Internet Explorer, Mozilla Firefox, Netscape Navigator, etc., et en ayant au préalable téléchargé des plug-in parce qu'un navigateur

seul n'est pas capable d'exploiter toutes les possibilités du Web ; l'installation de plugs-in additionnels est donc parfois nécessaire. Les plugs-in (ou modules externes) sont de petits programmes qui s'additionnent comme des extensions au programme original pour en élargir les capacités. Lorsque le navigateur rencontre un fichier sur Internet, il sait automatiquement qu'il doit utiliser un plug-in, soit parce que la page Web contient une balise spécifique (<embed>), soit parce que le plug-in a été associé à un type MIME lors de son installation. Rappelons qu'un type MIME (Multipurpose Independent Mail Extension) est un standard qui a été conçu il y a plusieurs années déjà pour gérer les fichiers attachés au courrier électronique. Par la suite, ce standard a été exploité par les navigateurs pour déterminer l'opération à exécuter lorsqu'ils téléchargent certaines pages sur Internet. Un type MIME est défini par un nom, une description, l'extension du fichier qu'il gère, l'application qui exécute ce type de fichier. Les principaux noms MIME sont : application, audio, image, texte, video.

- soit en utilisant des applications tel que Real Player, QuickTime, et tant d'autres.

Notre contribution est que, dans notre travail, nous avons spécialement développé WebTVPlayer 1.0 comme application cliente.

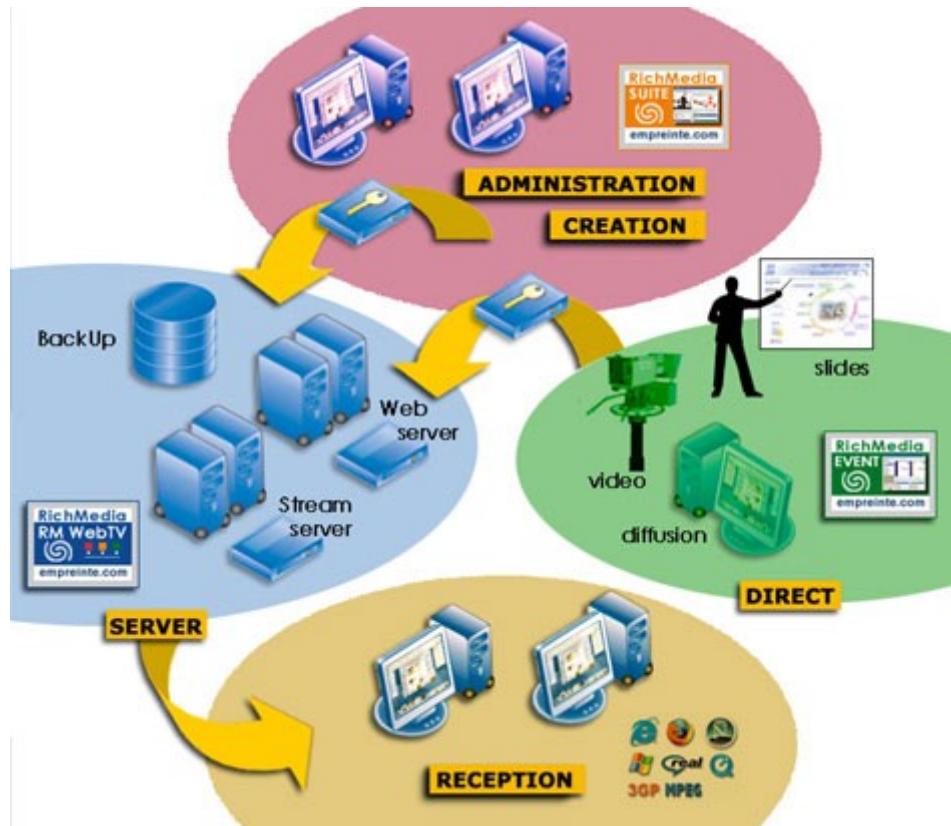


Figure 3.01: architecture d'une WEB TV.

### **3.3 Les exigences de la télévision sur Internet [6]**

-Les vidéos transmises lors d'une diffusion de télévision sur Internet sont très volumineuses et exigent des débits assez importants.

-Les techniques de compression doivent conduire à une qualité satisfaisante tout en gardant constant le débit, de préférence minimale. A ce titre, le MPEG-4 AVC est une norme en cours de développement qui apparaît aujourd'hui comme la solution du futur pour la diffusion de la télévision sur les réseaux IP. En effet, les débits sont de l'ordre de 1,5 Mbps mais parviennent à offrir une qualité télévisuelle.

### **3.4 Les solutions [6]**

Pour diffuser de la télévision sur Internet, deux solutions sont possibles. Compte tenu des capacités encore limitées d'Internet qui ne permet pas des diffusions avec une qualité suffisante, il faut, soit réduire le débit (et donc la qualité) des séquences vidéo en conservant le temps réel, soit s'affranchir de la diffusion en temps réel.

- La première solution, radicale, consiste à réduire la quantité d'information transmise - c'est à dire à diminuer la qualité des images et/ou leur fréquence - à proportion du débit numérique maximum permis par le ou les réseaux d'accès qui seront utilisés. On parle de Streaming (lecture vidéo en continu), détaillé plus loin.
- La seconde, plus riche de possibilités, consiste à s'affranchir du temps réel et met en oeuvre des techniques de téléchargement. En fait, on triche soit sur la qualité, soit sur le temps.

Dans le premier cas, la réduction de débit va s'effectuer en diminuant le poids des images : taux de compression plus importants, réduction des dimensions des fenêtres. La cadence des images pourra être inférieure à celle qui permet d'assurer une restitution fluide du mouvement (25 images par seconde traditionnellement pour la télévision). Dans ce scénario, l'image s'affiche au fur et à mesure de la réception. Dans les cas les moins favorables, lorsque les conditions de transmission sur le réseau se détériorent et sont telles que le débit devient insuffisant, l'image peut même se geler. La qualité est certes dégradée par rapport à une image de télévision, mais elle s'effectue en temps réel.

Dans le deuxième cas, on part d'une approche inversée par rapport à la précédente, et joue du temps au lieu de tricher sur la qualité. Elle repose sur le principe du téléchargement

préalable et se déroule en deux étapes. Préalablement au visionnement, la totalité des données vidéo est d'abord rapatriée sur le disque dur du terminal utilisateur.

Une fois les données stockées, l'ordinateur peut alors lire la séquence vidéo tout en conservant la cadence requise garantissant de ce fait une qualité constante et bien supérieure à la méthode précédente. Dans certains cas, il est possible de commencer à lire le fichier avant qu'il ne soit complètement téléchargé (Progressive Streaming) du moins pour ce qui concerne les données déjà téléchargées et à la condition bien sûr, que l'opération de téléchargement ne soit pas moins rapide que la lecture. La qualité des images reste optimale car contrairement à la méthode précédente, elle n'est pas altérée par les conditions de transfert sur le réseau. La qualité finale ne dépend plus du réseau, mais de la machine lectrice. La durée du téléchargement peut par contre être très variable et même parfois très longue car elle dépend du mode d'accès utilisé, de la taille du fichier (plus la qualité sera grande, plus le fichier sera volumineux) et des conditions de trafic sur Internet. On peut, bien entendu, jouer sur la taille des fichiers et donc sur la qualité des vidéos en utilisant les encodeurs propres au Streaming et en adaptant les paramètres d'encodage (taille des images, choix du débit, choix des codecs...). Des images en plein écran sont tout à fait possibles et c'est d'ailleurs ce qui est proposé sur les sites offrant des services de vidéo à la demande.

Chacune des deux méthodes a ses avantages et ses inconvénients : transmission en temps réel mais qualité réduite des images pour l'un, durée plus importante du transfert mais bonne qualité pour l'autre. Les différences entre Streaming et téléchargement s'évaluent également en terme de dimensionnement des flux, de serveurs spécifiques ou non, et de type de fichiers. Si le Streaming s'appuie sur des formats particuliers, il n'en n'est pas de même pour le téléchargement qui peut être réalisé pour tout type de fichier (pour la vidéo, AVI, Divx, MPEG-2, MPEG-4... et d'autres). Le *Tableau 3.01* résume ces deux types de solutions :

	<i>Téléchargement</i>	<i>Streaming</i>
<i>Diffusion en différé</i>	Oui	Oui
<i>Diffusion en direct</i>	Non	Oui
<i>Lecture simultanée</i>	Oui	Oui
<i>Protocoles de transfert</i>	Sans pertes	Pertes possibles mais plus rapide
<i>Passage des fire-walls</i>	Oui	Adaptation nécessaire
<i>Serveur</i>	Classique	Spécifique

*Tableau 3.01* : un tableau comparatif entre les deux solutions.

Notons que le Streaming soulève des difficultés sur les réseaux équipés de firewalls.

### **3.5 Le Buffering**

Le buffer est une mémoire tampon qui se remplit à un rythme très variable en fonction des fluctuations de débit sur le réseau mais qui est simultanément lisible à un rythme fixe. Son rôle est donc de lisser les irrégularités de débit sur le réseau : les variations, les ralentissements, les pertes de paquets... ne seront pas perçus s'ils ne sont pas importants.

Cette notion de buffer, si elle présente l'énorme avantage de fluidifier l'affichage des images, présente toutefois l'inconvénient d'introduire un retard au démarrage du visionnement : Il faut attendre quelques secondes que le buffer se remplisse avant de pouvoir commencer à visualiser un flux vidéo. Cet inconvénient est aujourd'hui minimisé avec le développement des accès internet à haut débit qui permettent d'utiliser l'excédent de capacité du réseau par rapport au débit réellement nécessaire au média à diffuser pour remplir ce buffer plus rapidement que le temps réel et donc de diminuer ce temps d'attente.

### **3.6 Les serveurs caches**

Pour pallier les problèmes de bande passante sur Internet, il a été aussi envisagé de rapprocher les données des utilisateurs, c'est à dire de déporter et de dupliquer les informations sur des serveurs caches répartis dans la zone géographique à couvrir. Plus il y aura de serveurs et moins les risques de congestion seront grands. Ces équipements pourront être alimentés par des réseaux dédiés, indépendant d'Internet. Des technologies spécifiques doivent être mises en oeuvre pour assurer leur gestion : mise à jour des documents stockés, aiguillage vers d'autres serveurs lorsque le serveur ne peut pas satisfaire une demande ou en fonction de l'encombrement du réseau. On parle de Content Delivery Network (CDN) pour caractériser ce type d'infrastructure. La société AKAMAI, leader sur ce marché, dispose de 18 000 serveurs répartis sur 70 pays.

### **3.7 Le Streaming : [7] [19] [20]**

Bien que deux solutions ont été proposées, le Streaming est évidemment la solution à prendre pour une transmission en temps réel de télévision en direct par exemple, mais la deuxième solution est acceptable pour la visualisation de programmes télévisés pré-enregistrées par exemple.

Ce qui nous intéresse surtout ici c'est la WEB TV dans son aspect où elle diffuse des émissions télévisées en direct sur Internet.

### ***3.7.1. Définition***

Le Streaming est une technique de transfert de données sous forme d'un flux régulier et continu qui permet de diffuser des contenus multimédias sur Internet, à la demande et en temps réel, sans solliciter le disque dur de l'utilisateur.

### ***3.7.2. Les buts du Streaming***

Le Streaming a pour buts de :

- permettre la diffusion de flux audio et vidéo sur Internet et réseaux locaux
- effectuer de la vidéo ou de l'audio sur demande (VoD : Video On Demand)
- diffuser en live (ex : une chaîne de télévision diffuse en permanence ses programmes sur Internet).

### ***3.7.3. Le principe du Streaming***

En temps normal pour consulter un fichier sur Internet, il faut l'avoir préalablement téléchargé sur son ordinateur. Lorsqu'il s'agit de fichiers de petites tailles (de l'ordre du kilo octet), cela ne pose aucun problème. Cependant, pour des fichiers plus importants le temps de téléchargement peut parfois être très long.

Le Streaming permet de remédier à ce genre de problèmes. En effet il permet le téléchargement et l'écoute simultanée de fichiers multimédia sans avoir à effectuer de copies locales. On peut donc transmettre des flux audio et vidéo sur le Web sans avoir à les télécharger.

C'est le *principe* que représente le Streaming.

### ***3.7.4. Le fonctionnement du Streaming***

Le Streaming consiste à découper les données en paquets dont la taille est adaptée à la bande passante disponible entre le Client et le Serveur.

Il utilise des formats de compression adaptés et permettant de distribuer et de diffuser en temps réel des contenus multimédias volumineux.

Le Streaming opère ainsi :

- la compression du fichier dans un format adapté pour le Web.
- l'échange de données selon un modèle client/serveur.

Ses principaux avantages résident dans son accès quasi-instantané, une transmission de durée illimitée, la possibilité de retransmission en direct, son adaptabilité à l'état du réseau et l'indexation des séquences.

Le Streaming consiste donc à diffuser des sources multimédia d'un serveur vers un client à travers un réseau IP. Et afin de minimiser leur taille et de permettre leur lecture en temps réel, elles doivent être compressées dans un format spécifique : c'est l'*encodage*. Le serveur segmente le fichier encodé en paquets susceptibles d'être diffusés sur le réseau.

Le serveur envoie le fichier vidéo ou audio par paquets de données qui seront traités par l'ordinateur de l'utilisateur au fur et à mesure de leurs arrivées. A cause des fluctuations réseaux les paquets n'arrivent pas toujours dans le bon ordre. On utilise donc une mémoire tampon pour regrouper les paquets dans le bon ordre. Cette mémoire tampon ou buffer est créée par le lecteur média de l'ordinateur de l'utilisateur. Au bout de quelques secondes, une fois que le buffer de réception possède assez d'informations, la lecture du flux commence et les images ou le son sont retransmis. La mémoire tampon a donc pour rôle de fluidifier le flux. Si la connexion réseau est mauvaise l'arrivée des paquets sera ralentie. Lorsque le buffer de réception est vide, la lecture s'arrête et reprendra lorsqu'elle possèdera assez de données pour continuer. L'image est alors figée.

Au niveau du client, les paquets seront assemblés afin de reconstituer le média. Il y a décodage en temps réel au niveau de l'utilisateur, à l'aide d'un *codec*. Afin d'harmoniser le débit, les données passent dans une mémoire tampon, ou buffer, avant d'être diffusé par le lecteur.

La qualité du Streaming dépend de deux facteurs :

- la bande passante,
- les logiciels d'encodage/décodage (*codec*) et les logiciels de lecture.

### **3.7.5. Les différents types de Streaming**

Un flux de streaming peut être diffusé de deux façons :

- de manière individuelle (*unicast*) : On effectue une connexion de point à point entre le serveur vidéo et l'utilisateur. L'utilisateur peut alors contrôler le flux et effectuer la lecture comme il le souhaite. Cette méthode demande beaucoup de ressources (bande

passante), car il faut allouer un flux unique par utilisateur. C'est pourtant l'utilisation la plus répandue aujourd'hui.

- à diffusion multiple (*multicast*) : On effectue cette fois ci une connexion de point à multi-points entre le serveur et les différents utilisateurs. En effet le serveur de vidéo envoie ici un seul flux qui est ensuite distribué à plusieurs utilisateurs. Cela nécessite l'utilisation de dispositifs permettant la distribution en multicast. L'utilisateur quant à lui ne peut pas piloter le flux puisqu'il est partagé avec plusieurs utilisateurs. Le multicast est typiquement utilisé pour la diffusion en live.

### 3.7.5.1. Le Streaming Unicast

L'unicast est caractérisé par des liaisons point à point entre une source unique et une destination unique. C'est le plus simple et le plus utilisé.

Etant donné que chaque ordinateur possède chacun une adresse IP pour son identification, chacun peut spécifier l'adresse IP de l'ordinateur à qui il voudra communiquer. Les éléments du réseau (commutateurs, routeurs, ...) dirigent l'information dans la bonne direction pour que les données puissent arriver au bon endroit. Seule la machine ayant l'adresse indiquée dans la requête peut recevoir et traiter l'information.

Le client contacte le serveur de Streaming grâce au protocole RTSP. En réponse à cette requête, le serveur retourne via RTSP, une description de la session de Streaming qu'il va ouvrir. Une session de Streaming est composée d'un ou plusieurs flux, par exemple audio ou vidéo. Le serveur informe le client du nombre de flux. Il donne aussi des informations décrivant les flux comme le type du média et le codec de compression. Les flux sont quant à eux diffusés séparément via le protocole RTP.

Un point fort du Streaming unicast est que chaque client peut contrôler les données multimédias mise à sa disposition avec les méthodes PLAY, PAUSE, STOP, TEARDOWN (cf Chapitre 5) et peut choisir le débit pour sa connexion.

Sur la *Figure 3.02*, les clients C1, C2, C3, C4 contactent le Serveur S, et ce dernier se voit obligé de répéter l'envoi de chaque information à chaque destinataire (liaison point à point). Cependant, la même information va être transmise de nombreuses fois par les routeurs (autant de fois qu'il y a de destinataires différents). Elle est une grande consommatrice de bande passante.

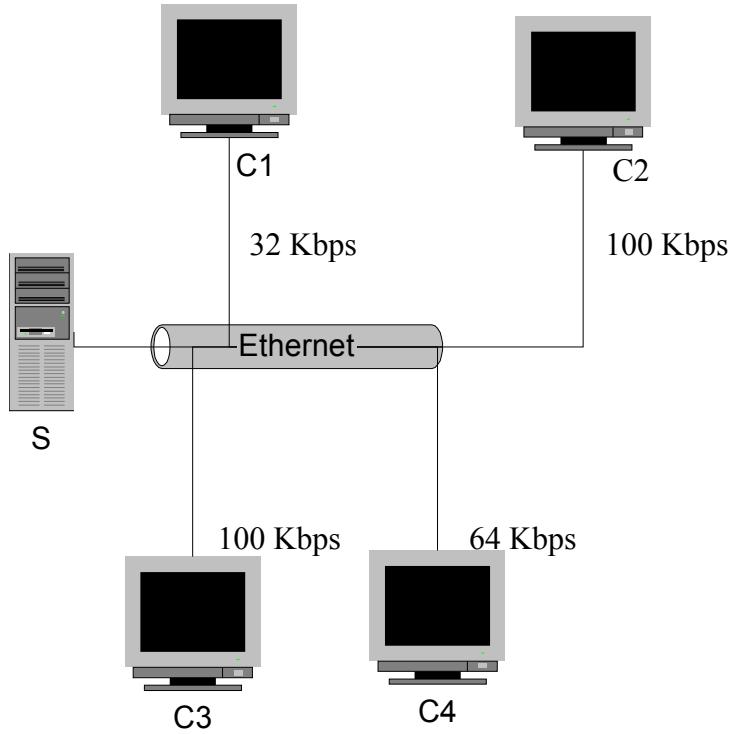


Figure 3.02: le streaming unicast

### 3.7.5.2. Le Streaming Multicast

La technique multicast ou multidiffusion est un procédé de routage qui consiste à envoyer un même paquet de données à plusieurs destinataires simultanément. Il permet l'envoi des datagrammes à plusieurs destinations de façon performante. IP utilise les adresses de classe D pour indiquer qu'il s'agit d'un envoi multipoint.

Les groupes de multidiffusion sont dynamiques : un hôte peut se rattacher à un groupe et le quitter à tout moment, l'hôte devant être capable d'émettre et de recevoir des datagrammes en multicast.

En streaming multicast, une seule copie de chaque flux est envoyée à chaque branche du réseau, ce qui permet de réduire considérablement le trafic lors d'une diffusion pour de nombreux clients. Cependant, les clients doivent s'abonner au groupe pour recevoir le flux.

Lors d'une diffusion en multicast, un seul flux est émis pour tous les utilisateurs. Cette technique a l'avantage de diminuer la bande passante nécessaire et de diminuer la charge en niveau du serveur. En effet, il n'y a plus à gérer plusieurs connexions distinctes.

Sur la *Figure 3.03*, le Serveur n'aura pas à répéter l'information à chaque destinataire. C'est le routeur, supportant la technique multicast, qui se charge de dupliquer ces

informations. Les clients C1, C2, C3, C4 reçoivent le même flux au même instant. La charge du réseau en est considérablement allégée et l'économie de bande passante est substantielle.

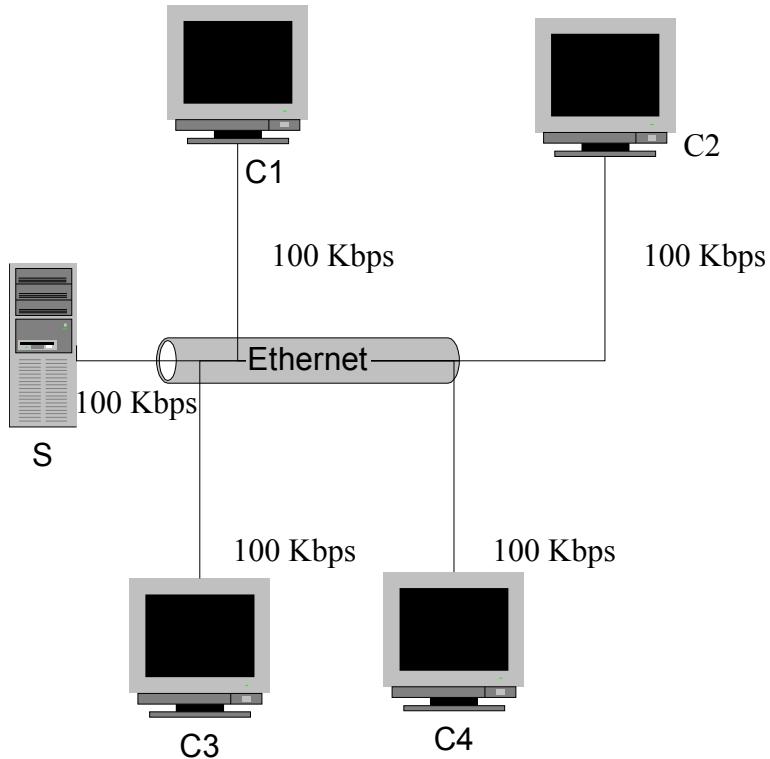


Figure 3.03 : le streaming multicast

### 3.7.6. Les solutions en Streaming

Il y a trois composants principaux qui sont mis en jeu durant une session de Streaming :

- le lecteur ou Player (côté client),
- le serveur Streaming
- l'encodeur streaming (côté serveur).

Il existe aujourd’hui trois grands acteurs qui se partagent le marché du Vidéo-Streaming :

- Real Networks, dont les plateformes supportées par le Player sont : Windows, Linux, Solaris, Symbian, Unix, MacOS.
- Apple dont le Player supporte les plateformes : MacOS, Windows.
- Microsoft, qui supporte les plateformes Windows, MacOS. Il existe des clients Linux susceptibles de recevoir des flux au format Windows Media.

### ***3.7.7. Les points forts et les points faibles du Streaming***

Les avantages de l'utilisation du Streaming sont :

- Aucune contrainte d'espace ni de temps pour une communication planétaire,
- Les internautes aux quatre coins du monde peuvent visualiser le même programme,
- Les internautes peuvent choisir des programmes à la carte qui les concernent directement,
- Un impact optimisé grâce à une communication ciblée : proposer des contenus ciblés à un public déterminé.
- Multiple choix de connexion,
- Valorisation d'un site web et augmentation du trafic.

Mais elle présente quelques points faibles :

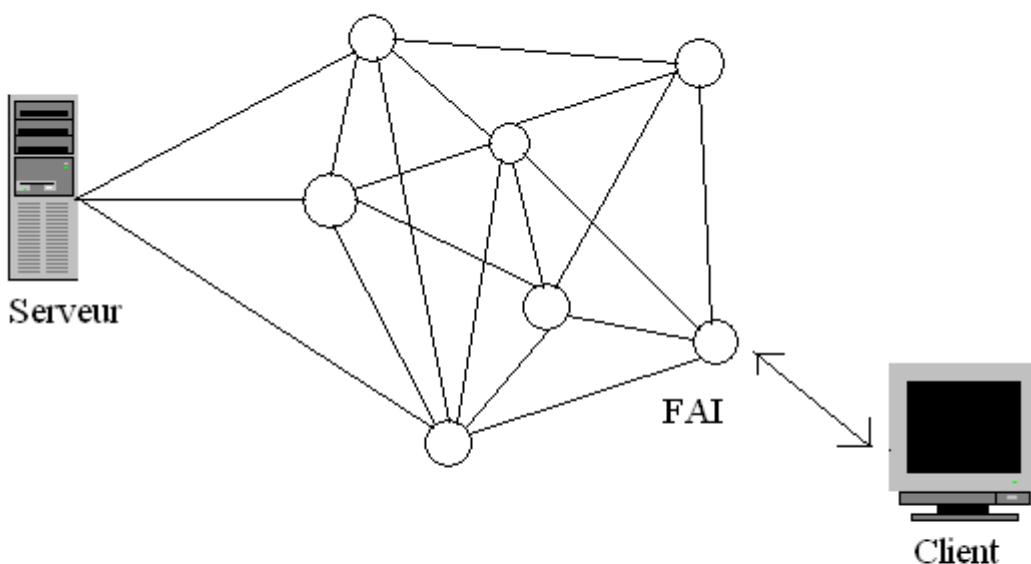
- La qualité sonore et vidéo dépend du trafic du réseau.
- On a besoin d'un serveur Streaming au lieu du serveur web habituel.

## CHAPITRE 4 : LES TECHNOLOGIES D'ACCES

### 4.1 Historique

Internet est né aux USA au moment de la guerre froide. C'est un réseau dont le concept d'acheminement des messages a été marqué à l'origine par des préoccupations militaires. L'idée qui a prévalu au moment de sa naissance était que les informations devaient pouvoir être acheminées quelles que soient les destructions opérées sur les réseaux de transmission. D'où l'idée de réaliser un support de communication comprenant un maillage très serré et de multiples imbrications de tous les réseaux entre eux : il existait ainsi plusieurs chemins pour aller d'un point à un autre et si l'un d'entre eux était coupé, les données pouvaient automatiquement en prendre un autre.

Aujourd'hui, Internet est un vaste réseau né du regroupement à l'échelle mondiale d'une multitude de réseaux aux caractéristiques très variables, mais la philosophie générale qui prévalait à l'origine a été conservée. Du fait de ce maillage important, voir *Figure 4.01*, ce réseau offre une grande souplesse de fonctionnement.



*Figure 4.01* : le maillage important d'Internet : pour les données, de multiples chemins sont possibles entre l'ordinateur de l'utilisateur et le site Web demandé.

## 4.2 Des accès hétérogènes

### 4.2.1. L'ADSL (Asymmetric Digital Subscriber Line)

L'ADSL est aujourd'hui la principale technologie pour l'accès à Internet à haut débit.

Compte tenu de son importance, nous accorderons une place spécifique à sa description et à ses déclinaisons possibles.

L'ADSL ouvre l'accès à des dizaines de programmes télé, sans parabole ni la moindre antenne raccordée. C'est une technologie qui s'appuie sur le réseau téléphonique traditionnel. Elle n'est que l'un des membres de la grande famille de normes DSL (Digital Suscriber Line ou Ligne d'abonné numérique). Ces technologies ne s'appliquent pas sur tout le réseau de transmission téléphonique, mais uniquement sur ce que l'on appelle la boucle locale, c'est-à-dire *le dernier kilomètre* du réseau téléphonique qui relie chaque abonné au central téléphonique. Ces liaisons sont constituées par les simples paires de fils de cuivre du téléphone.

Contrairement aux communications téléphoniques, c'est bien de transfert de données numériques dont il s'agit ici. L'ADSL permet une connexion Internet permanente et indépendante de l'utilisation du téléphone analogique classique.

La bande passante d'une ligne téléphonique constituée de deux fils en cuivre est de plusieurs Mhz mais elle est très variable en fonction de la distance. Jusqu'à l'avènement de l'ADSL, les services de téléphonie n'utilisaient que les fréquences les plus basses, inférieures à 4 KHz. Les technologies ADSL exploitent les fréquences supérieures, jusqu'à 1,1MHz pour l'ADSL puis ensuite jusqu'à 2,2 MHz pour la norme ADSL2+. Elles sont réparties (hormis la bande réservée à la téléphonie classique) en deux bandes distinctes :

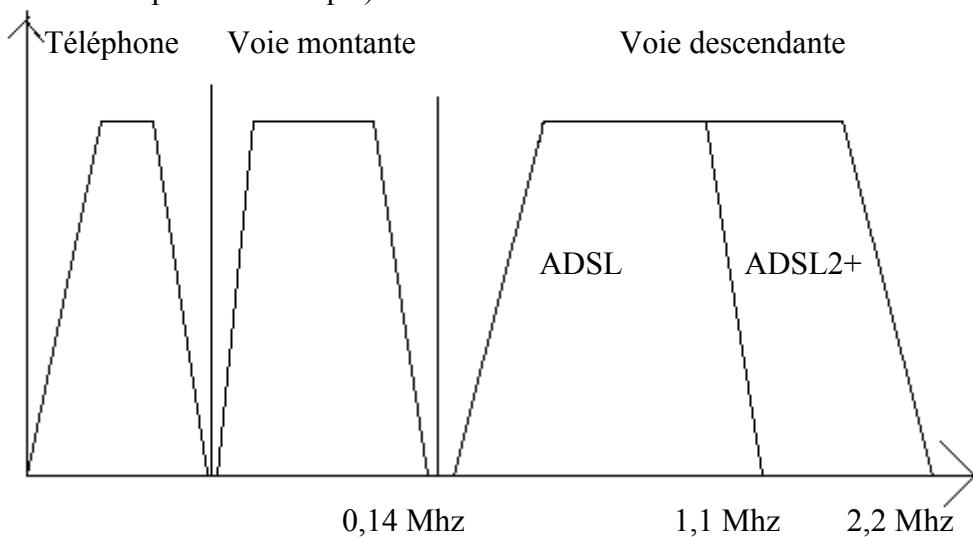


Figure 4.02 : spectre de fréquences de l'ADSL

la bande inférieure pour la voie montante (pour les données numériques émises par le client) et la bande supérieure descendante beaucoup plus étendue et donc offrant un débit plus important.

Les débits proposés par l'ADSL ne sont pas donc identiques dans les deux sens de circulation d'où la lettre A signifiant Asymétrique dans l'acronyme ADSL. Ils ne dépassent pas 1 Mb/s dans le sens montant. Dans le sens descendant (c'est-à-dire vers l'usager), ils sont aux maximum de 8 Mb/s avec l'ADSL mais peuvent atteindre 25 Mb/s avec l'ADSL2+ qui a été normalisé il y a deux ans.

L'ADSL2+ ne peut pas être proposé à l'ensemble de la population parce que ses performances décroissent très rapidement avec la distance. Avec l'ADSL le débit maximum théorique de 8 Mb/s ne pourra être atteint que dans le secteur proche du central téléphonique. À 2,5 km de distance, il ne sera plus que de 4 Mb/s environ.

D'autres éléments intrinsèques peuvent également intervenir dans cette limitation : la qualité la ligne, le diamètre des fils, les interférences avec les fils voisins... Avec l'ADSL2+, c'est le même problème. S'il autorise des débits pouvant atteindre 25 Mb/s pour les abonnés les plus proches du central, ses performances décroissent d'autant plus rapidement avec la distance que les fréquences mises en jeux sont deux fois plus élevées que pour l'ADSL classique : en conséquence, le gain en débit apporté par la nouvelle norme ne sera significatif que dans un rayon de 2,5 km environ autour du central. Au delà, le gain sera minime par rapport à l'ADSL classique, les clients les plus éloignés ne percevront pas de différence.

#### 4.2.1.1. Se connecter à Internet via l'ADSL

##### 4.2.1.1.1. *La boucle locale*

La ligne en cuivre est l'élément le plus ancien du dispositif, parfois en service depuis de nombreuses décennies pour les services téléphonique traditionnels.

L'ouverture de ce réseau aux technologies ADSL, si elle ne nécessite pas en général de modification particulière des lignes (sauf si elles sont de mauvaise qualité), implique cependant l'implémentation d'équipements spécifiques aux deux extrémités de la ligne de cuivre.

#### *4.2.1.1.2. Le dégroupage*

Pour les fournisseurs d'accès à Internet concurrents, il n'est pas économiquement possible de développer leur propre réseau final de distribution vers les usagers. Il a donc été décidé par voie législative que l'opérateur historique devait permettre l'accès à la boucle locale à ses concurrents : c'est le dégroupage. A charge ensuite à ces nouveaux opérateurs d'installer leurs propres équipements dans les centraux téléphoniques et de déployer leurs propres réseaux à hauts débits.

On distingue :

- le dégroupage total : l'opérateur alternatif loue l'utilisation de la boucle locale et gère la totalité des services (y compris la téléphonie). Il a alors la totale maîtrise sur l'offre qu'il propose.
- le dégroupage partiel : l'opérateur téléphonique fixe conserve la gestion des communications téléphoniques analogiques. L'opérateur alternatif gère les deux bandes de fréquences hautes nécessaires à l'ADSL et assure la fourniture des services liés à Internet et les autres nouveaux services (téléphonie sous IP, télévision, VOD...).
- l'accès sans dégroupage : l'opérateur de téléphonie fixe assure la gestion des communications téléphoniques analogiques et prend totalement en charge la collecte et le transport des données Internet jusqu'au point d'accès du réseau du FAI.

#### *4.2.1.1.3. Les éléments de base d'une connexion Internet via l'ADSL*

Les équipements propres à l'ADSL sont de deux ordres.

Côté utilisateur, on trouvera un modem spécifique qui sera inséré entre la prise téléphonique et le micro-ordinateur. La liaison entre le modem et le micro-ordinateur pourra être de type filaire (USB ou Ethernet), hertzienne (Wifi) ou par le réseau électrique (CPL).

Côté opérateur, le DSLAM (Digital Subscriber Line Access Multiplexer) regroupe les données numériques des abonnés et les multiplexe sur un seul lien à haut débit vers le coeur du réseau. Cette interface est installée dans les centraux téléphoniques. Suivant le degré de dégroupage adopté, ces équipements pourront être ceux de l'opérateur fixe ou ceux de tout opérateur alternatif que l'opérateur fixe a obligation d'accueillir dans ses locaux techniques.

A l'autre extrémité de la chaîne, le BAS ou Broadband Access Server réalise l'interface entre le réseau de collecte et les réseaux d'accès aux fournisseurs d'accès Internet pour la fourniture des contenus. Cet élément est notamment chargé de l'authentification des

utilisateurs et du routage de leurs données vers le fournisseur d'accès concerné.

Les filtres placés sur le connecteur téléphonique au domicile de l'usager ne doivent pas être oubliés. Ils assurent la séparation entre les fréquences les plus basses destinées au téléphone analogique et les fréquences plus élevées nécessaires au fonctionnement de l'ADSL, si la ligne n'est pas totalement dégroupée. Si la ligne est totalement dégroupée, ces filtres ne sont plus nécessaires, puisque ne circulent plus sur la ligne que les fréquences hautes liées à l'ADSL.

La *Figure 4.03* résume ces explications.

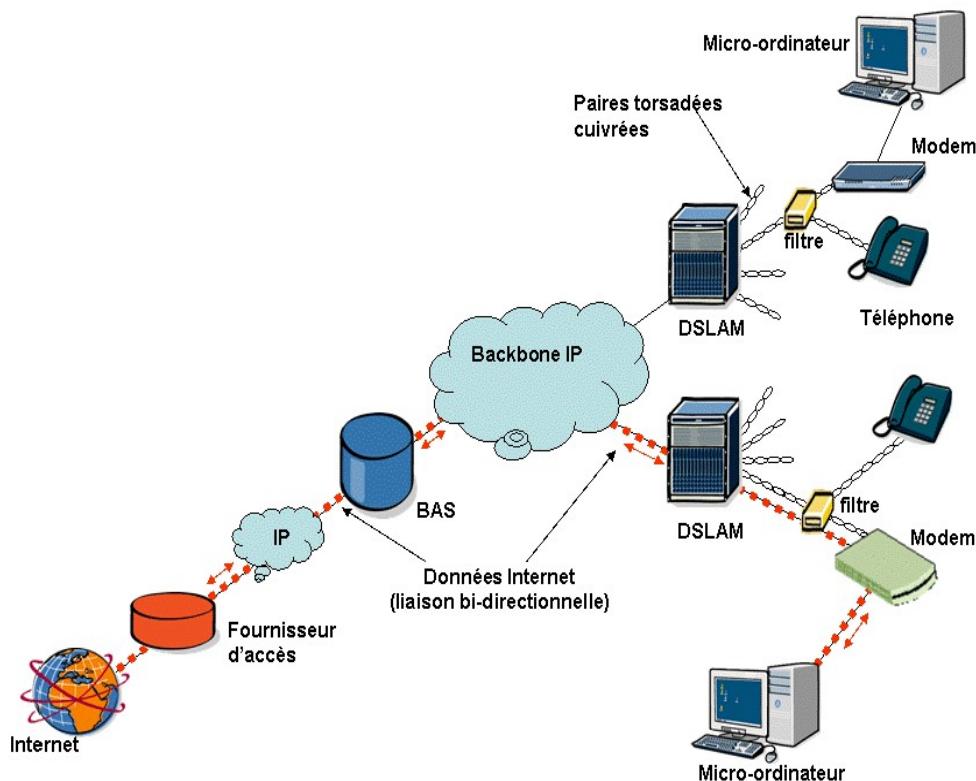


Figure 4.03 : schéma général d'une connexion Internet via l'ADSL

#### 4.2.2. Les autres technologies d'accès

Le câble est une alternative à l'ADSL, mais qui reste limitée aux agglomérations câblées. Initialement destinés à la diffusion des programmes de télévision, en analogique dans un premier temps puis en numérique aujourd'hui, les réseaux câblés urbains permettent aussi la transmission bilatérale de données à haut débit. Les offres commerciales des câblo-opérateurs sont du même ordre que pour l'ADSL (jusqu'à environ 20 Mb/s dans le sens descendant). Le débit peut être très variable selon le nombre d'utilisateurs simultanés. Tout

comme pour l'ADSL, la connexion à Internet est permanente.

La technologie ADSL et le câble ne permettent pas de couvrir la totalité d'un territoire. Des solutions alternatives existent pour les zones plus désertiques, hors agglomérations câblées ou situées trop loin d'un répartiteur pour l'ADSL. Elles s'appuient sur le satellite, les ondes radio ou sur le réseau de distribution de l'électricité.

Largement utilisé pour la télévision, le satellite apporte une réponse pour la transmission de données ou pour l'accès individuel à Internet avec des débits qui pourraient être importants, mais qui sont généralement limités commercialement à quelques Mb/s pour ce qui concerne les particuliers. Un équipement spécifique (carte de réception à la norme DVB) doit être intégré au micro-ordinateur et être relié à une antenne parabolique extérieure. Deux méthodes sont proposées pour la voie montante (requêtes des utilisateurs). La première s'appuie sur la ligne téléphonique classique par l'intermédiaire d'un modem. La seconde, réservée aux professionnels (collectivités territoriales, entreprises...) est plus onéreuse et s'appuie exclusivement sur le satellite : les liaisons sont alors bi-directionnelles, la parabole est également utilisée en émission.

Le Wimax (IEEE 802.16) apportera également une réponse, hertzienne en l'occurrence (bande de 3,5 Ghz), pour s'affranchir des derniers kilomètres en direction des utilisateurs. Cette technologie radio, pas encore totalement stabilisée sur le plan de la normalisation, présente de nombreux avantages : des liaisons de quelques dizaines de kilomètres, des débits élevés et bidirectionnels, une installation facile, une antenne de dimension réduite.

Une autre solution alternative s'appuie sur le réseau électrique. L'idée de superposer des signaux numériques au courant électrique n'est pas récente mais elle trouve ici une application nouvelle avec le développement de l'Internet. Si elle reste encore expérimentale pour le câblage des agglomérations, elle est par contre totalement opérationnelle à l'intérieur, en remplacement des réseaux filaires d'établissements. En extérieur, la boucle locale électrique autorise des débits élevés (théoriquement plusieurs dizaines de Mbps, pratiquement beaucoup moins) mais présente aussi des contraintes, notamment, en terme de distance utile puisque qu'elles ne peuvent pas excéder quelques centaines de mètres (300m) sauf à installer des répéteurs.

La BLR (Boucle Locale Radio) est un réseau de raccordement radioélectrique d'abonnés ouverts au public notamment dans la bande 3,4 – 3,8 Ghz.

Il y a aussi les nouveaux opérateurs qui déplacent de la fibre optique.

Bien sûr, cette liste n'est pas complète car plusieurs technologies d'accès sont développées au fur et à mesure ;

Toutefois, l'ADSL reste le réseau d'accès le plus utilisé actuellement en terme de Télévision sur Internet.

## **CHAPITRE 5 : LES PROTOCOLES UTILISES**

### **5.1 Définition [8] [9]**

Un protocole est un ensemble de règles permettant à plusieurs ordinateurs de dialoguer entre eux. Le protocole principal de communication permettant de transmettre des données sur Internet est le protocole TCP/IP qui est une pile de protocoles. En effet, le terme TCP/IP n'est pas limité à l'expression Transmission Control Protocol /Internet Protocol, mais il recouvre aussi toute une famille de protocoles comme UDP, protocole sans connexion, FTP, HTTP, Telnet, et tant d'autres.

### **5.2 Généralités sur les protocoles utilisés en Streaming**

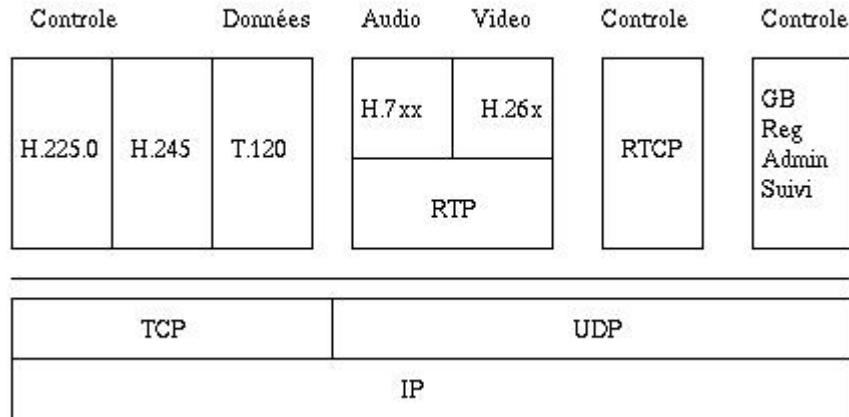
Les protocoles utilisés pour des données statiques ne sont pas adéquats pour supporter le Streaming. Par exemple FTP et HTTP sont basés sur TCP (Transmission Control Protocol) utilisé dans le cadre d'une communication fiable (contrôle d'erreur, retransmission si perte, perte de débit). Or tous ces contrôles de données ne nous sont pas utiles pour effectuer du Streaming et occasionne une perte de débit, essentiel au bon fonctionnement de ce processus.

C'est pourquoi le Streaming se base le plus souvent sur le protocole de transport UDP (User Datagram Protocol). Ce protocole ne garantit pas l'ordre de réception des paquets ni leur arrivée à destination mais nous permet d'obtenir un débit optimal. TCP et UDP sont des protocoles de transport autour desquels la plupart des applications fonctionnent aujourd'hui.

Le standard Internet pour transporter des données média en temps réel est RTP qui est un protocole de plus haut niveau.

RTP nous permet d'identifier le type de données transmises, détermine l'ordre dans lequel les paquets doivent être interprétés et synchronise les données média à partir de différentes sources. Les paquets RTP ne sont en aucun cas garantis d'arriver dans l'ordre ou d'arriver tout simplement. C'est au receveur de reconstruire la séquence de paquets et de détecter les paquets perdus à partir de l'entête RTP. Comme RTP ne fournit aucun mécanisme de contrôle ni aucune garantie de la qualité de service, il existe un protocole de contrôle RTCP qui permet de gérer la qualité des données transmises. Il fournit également du contrôle et un mécanisme d'identification pour les transmissions RTP.

Le protocole RTSP est une amélioration du protocole RTP. Il contrôle entre autre les propriétés de temps d'une vidéo. La *Figure 5.01* montre une vue d'ensemble de ces protocoles.



*Figure 5.01: une vue d'ensemble des protocoles.*

Le protocole de premier niveau, IP, s'occupe du routage des informations entre l'expéditeur et le destinataire : il accomplit sa tâche en divisant les informations en paquets (de 1500 octets) et leur adjoint une adresse de provenance et de destination.

TCP s'appuie sur IP pour gérer le transfert de données entre l'expéditeur et le destinataire. TCP fournit également les mécanismes permettant d'établir les connexions, de vérifier l'arrivée dans le bon ordre des données, de gérer des données perdues, les erreurs et de récupérer des données concernées.

Ces protocoles seront détaillés plus loin.

### 5.3 Le modèle de référence OSI [8] [9]

Le nombre et la taille des réseaux sont considérables, et il arrive parfois qu'il y ait une incompatibilité entre ces réseaux et il devient difficile d'établir des communications entre des réseaux fondés sur des spécifications différentes.

Pour résoudre ce problème, l'ISO a examiné de nombreuses structures de réseau. Elle a reconnu l'opportunité de créer un modèle réseau qui aiderait les concepteurs à mettre en œuvre des réseaux capables de communiquer entre eux et de fonctionner afin d'assurer une interopérabilité. L'ISO a donc publié le modèle de référence OSI en 1984, voir *Figure 5.02*.

Le modèle de référence OSI comporte sept couches numérotées, chacune illustrant une fonction bien précise. Cette répartition des fonctions réseaux est appelée *organisation en couches*.

<b>7</b>	<b>Application</b>
<b>6</b>	<b>Présentation</b>
<b>5</b>	<b>Session</b>
<b>4</b>	<b>Transport</b>
<b>3</b>	<b>Réseau</b>
<b>2</b>	<b>Liaison de données</b>
<b>1</b>	<b>Physique</b>

*Figure 5.02 : le modèle OSI.*

Le *Tableau 5.01* affiche une brève description des fonctions de chaque couche :

<b>7</b>	<b>Application</b>	Fournit les protocoles et les fonctions nécessaires aux applications utilisateurs qui doivent accomplir des tâches de communication.
<b>6</b>	<b>Présentation</b>	Gère la représentation des données entre les systèmes.
<b>5</b>	<b>Session</b>	Gère les connexions entre les applications coopérantes.
<b>4</b>	<b>Transport</b>	Offre des services supplémentaires par rapport à la couche réseau, garantit l'intégrité des données.
<b>3</b>	<b>Réseau</b>	Gère les connexions entre les nœuds du réseau.
<b>2</b>	<b>Liaison de données</b>	Prend les données de la couche physique et fournit ses services à la couche réseau.
<b>1</b>	<b>Physique</b>	Transmet les bits à travers un canal de communication.

*Tableau 5. 01 : une description des sept couches du modèle OSI.*

Chaque couche du modèle OSI doit exécuter une série de fonctions pour que les paquets de données puissent circuler d'un ordinateur source vers un ordinateur de destination sur un réseau.

#### **5.4 L'architecture TCP/IP [8] [9]**

Même si le modèle OSI est universellement connu, historiquement et techniquement, la norme ouverte d'Internet est le protocole TCP/IP.

Dans les années 70, la défense américaine, devant le foisonnement de machines utilisant des protocoles de communication différents et incompatibles, décide de définir sa propre architecture. Cette architecture TCP/IP, est la source du réseau Internet ; elle est aussi adoptée par l'Intranet (application de l'Internet dans un réseau local d'entreprise).

Le modèle de référence TCP/IP, voir *Figure 5.03*, et la pile de protocole TCP/IP rendent possible l'échange de données entre deux ordinateurs, partout dans le monde.

L'architecture TCP/IP utilise moins de niveau que celle utilisée par le modèle OSI :

4	APPLICATION
3	TRANSPORT
2	INTERNET
1	ACCES RESEAU

*Figure 5.03: le modèle TCP/IP.*

Le modèle TCP/IP met l'accent sur une souplesse maximale, au niveau de la couche Application, à l'intention des développeurs de logiciels. La couche Transport fait appel à deux protocoles : le protocole TCP et le protocole UDP.

Dans le modèle TCP/IP, IP est le seul et unique protocole utilisé, et ce, quels que soient le protocole de transport utilisé et l'application qui demande des services réseaux. Il s'agit là d'un choix de conception délibéré. IP est un protocole universel qui permet à tout ordinateur de communiquer en tout temps et en tout lieu.

## 5.5 Le protocole IP

Le protocole IP est au cœur du fonctionnement d'Internet. Il assure, sans connexion, un service non fiable de délivrance de datagrammes IP. Le service est non fiable car il n'existe aucune garantie pour que les datagrammes IP arrivent à destination. Certains peuvent être perdus, dupliqués, retardés, altérés ou remis dans le désordre. Le mode de transmission est non connecté car IP traite chaque datagramme indépendamment de ceux qui le précèdent et le suivent. Ainsi, en théorie, au moins, deux datagrammes IP issus de la même machine et ayant la même destination peuvent ne pas suivre obligatoirement le même chemin. Le rôle du protocole IP est centré autour des trois fonctionnalités suivantes :

- définir le format du datagramme IP qui est l'unité de base des données circulant sur Internet.
- définir le routage dans Internet.
- définir la gestion de la remise non fiable des datagrammes.

La *Figure 5.04* illustre le format d'un datagramme IP. Un datagramme IP contient un en-tête et des données IP. Un message peut être transmis sous la forme d'une série de datagrammes qui sont ensuite rassemblés dans le message sur la station réceptrice.

VERS	HLEN	Type de service	Longueur totale
Identification		Indicateurs	Décalage de fragment
Durée de vie	Protocole	Somme de contrôle d'en-tête	
Adresse IP source			
Adresse IP destination			
Option IP	Remplissage		
Données			

*Figure 5.04 : le datagramme IP.*

Les champs du datagramme IP sont :

- *Version* : indique la version du protocole IP utilisée (champ de 4bits).
- *HLEN* : indique la longueur de l'en-tête du datagramme en mots de 32 bits (champ de 4bits).
- *Type de service* : indique l'importance qui lui a été accordée par un protocole de couche supérieure donné (champ de 8 bits)
- *Longueur totale* : précise la longueur du paquet IP en entier, y compris les données et l'en-tête, en octets (champ de 16 bits).
- *Identification* : contient un nombre entier qui identifie le datagramme actuel (champ de 16 bits).
- *Indicateurs* : Un champ de 3 bits dont les 2 bits inférieurs contrôlent la fragmentation (un bit précise si le paquet peut être fragmenté et le second bit indique si le paquet est le dernier fragment d'une série de paquets fragmentés) (champ de 3 bits).
- *Décalage de fragment* : ce champ sert à rassembler les fragments du datagramme (champ de 13 bits).
- *Durée de vie* : un compteur qui décroît graduellement, par décrémentation, jusqu'à zéro. A ce moment, le datagramme est supprimé, ce qui empêche les paquets d'être continuellement en boucle (champ de 8 bits).
- *Protocole* : précise le protocole de couche supérieure qui recevra les paquets entrants après la fin du traitement IP (champ de 8 bits).
- *Somme de contrôle d'en-tête* : assure l'intégrité de l'en-tête IP (champ de 16 bits).
- *Adresse IP source* : indique le nœud d'émetteur.
- *Adresse IP destination* : indique le nœud récepteur (champ de 32 bits).
- *Option IP* : cet élément permet au protocole IP de supporter différentes options, telles que la sécurité (champ de longueur variable).
- *Remplissage* : Des zéros sont ajoutés à ce champ pour s'assurer que l'en-tête IP est toujours un multiple de 32 bits.
- *Données* : cet élément contient les informations de couche supérieure (champ de longueur variable, maximum 64 Ko).

## 5.6 Le protocole TCP (Transmission Control Protocol)

TCP est un protocole qui permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle (c'est-à-dire le protocole IP). 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 flux 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.

Une trame TCP est constituée comme indiquée sur la *Figure 5.05* :

0	31
Port source	Port destination
Numéro de séquence	
HLEN	Réservé
Somme de contrôle	Bits de code Fenêtre
Options	Pointeur d'urgence Remplissage
Données	

*Figure 5.05* : le datagramme TCP.

Signification des différents champs :

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

numéro de séquence est le numéro de séquence initial utilisé pour synchroniser les numéros de séquence.

- *Numéro d'accusé de réception* : Dernier segment reçu par le récepteur (champ de 32 bits).
- *HLEN* : indique la longueur de l'en-tête du datagramme en mots de 32 bits (champ de 4 bits).
- *Réservé* : Champ inutilisé actuellement mais prévu pour l'avenir (champ de 6 bits).
- *Bits de code* : Les drapeaux représentent des informations supplémentaires (champ de 6 bits).
- *Fenêtre* : Champ permettant de connaître le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception (champ de 16 bits).
- *Somme de contrôle (checksum)* : 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* : indique le numéro d'ordre à partir duquel l'information devient urgente (champ de 16 bits).
- *Options* : Descriptions diverses (taille variable).
- *Remplissage* : On remplit l'espace restant après les options avec des zéros pour avoir une longueur de 32 bits.

## 5.7 Le protocole UDP (User Datagram Protocol)

Le protocole UDP 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.

L'en-tête du datagramme UDP est donc très simple et représentée sur la *Figure 5.06* :

0	31
Port source (16 bits)	Port destination (16 bits)
Longueur (16 bits)	Somme de contrôle (16 bits)
Données (longueur variable)	

*Figure 5.06* : le datagramme UDP.

Signification des différents champs :

- *Port source* : il s'agit du numéro de port correspondant à l'application émettrice du datagramme. 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.
- *Port destination* : Ce champ contient le port correspondant à l'application de la machine émettrice à laquelle on s'adresse.
- *Longueur* : ce champ précise la longueur totale du datagramme, en-tête comprise, or l'en-tête a une longueur de 8 bits donc le champ longueur est supérieur ou égal à 8.
- *Somme de contrôle* : Il s'agit d'une somme réalisée de telle façon à pouvoir contrôler l'intégrité du datagramme.

## 5.8 Les protocoles de la couche haute

Un ensemble de protocoles a été proposé pour le Streaming multimédia sur Internet. Les trois protocoles majeurs que nous allons étudier sont le protocole RTP, le protocole RTCP et le protocole RTSP.

### 5.8.1. Le protocole RTP (*Real-time Transport Protocol*)

#### 5.8.1.1. L'objet du protocole RTP

RTP est un protocole proposé par l'IETF (RFC 1889). RTP est utilisé par beaucoup de protocoles, tel que RTSP pour les applications en streaming, le H.323 et le SIP pour les applications de la téléphonie sur IP, etc...

Il fournit le format de la livraison des données pour tous ces protocoles.

#### 5.8.1.2. Les caractéristiques du protocole RTP

RTP fournit des fonctions de transport pour les applications temps réel (audio et vidéo) sur des services réseaux unicast ou multicast. RTP est au-dessus du transport UDP/TCP, mais pratiquement, il s'appuie sur UDP. De plus, RTP peut être véhiculé par des paquets multicast afin d'acheminer des données vers des destinataires multiples.

Le rôle principal de RTP consiste à mettre en œuvre des numéros de séquence de paquets IP pour la synchronisation en temps réel des données audio et vidéo.

Cependant, RTP n'apporte pas de fiabilité dans le réseau, ne fournit pas de contrôle de flux et n'offre pas une garantie dans le délai de livraison des données.

Le contrôle de flux est assuré par le protocole RTCP qui est désigné pour travailler conjointement avec RTP. Chacun d'eux utilise un port séparé d'une paire de ports. RTP utilise le port pair, et RTCP le port impair immédiatement supérieur.

#### 5.8.1.3. Les fonctions du protocole RTP

Les fonctions du protocole RTP sont :

- le séquencement des paquets par une numérotation.

Cette numérotation permet de détecter les paquets perdus, ce qui est important dans la reconstitution de la donnée multimédia. En effet, la perte d'un paquet n'est pas en soi un problème, s'il y en a trop de perdu. En revanche, repérer qu'un paquet a été perdu est impératif pour en tenir compte et éventuellement le remplacer par une synthèse déterminée en fonction des paquets précédents et suivants.

- l'identification de ce qui est transporté dans le message pour permettre, par exemple, une compensation en cas de perte.

- l'identification de trame.

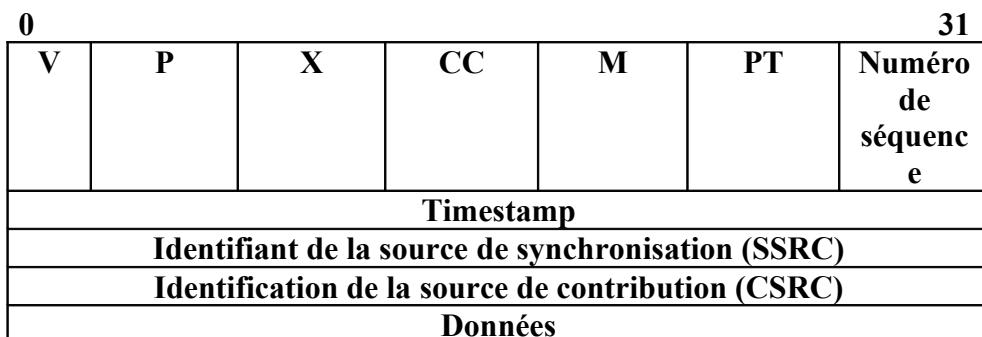
Les applications d'audio et de vidéo sont transportées dans des trames dont la dimension dépend des codecs effectuant la numérisation. Ces trames sont incluses dans les paquets pour être transportées et elles doivent être récupérées facilement au moment de la dépaquetisation pour que l'application soit décodée simplement.

- l'identification de la source.

En effet, dans les applications en multicast, l'identité de la source doit être déterminée.

#### 5.8.1.4. L'entête du protocole RTP

Le format du datagramme RTP est représenté sur la *Figure 5.07* :



*Figure 5.07* : le datagramme RTP.

Voici la signification des différents champs de l'en-tête :

- *Le champ Version V* : 2 bits, indique la version du protocole.
- *Le champ padding P* : 1 bit, si P est égal à 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 suivi d'un paquet d'extension.
- *Le champ CSRC count ou CC* : 4 bits, contient le nombre CSRC qui suit l'en-tête.
- *Le champ marker M* : 1 bit, son interprétation est définie par un profil d'application (profile).
- *Le champ Payload Type ou PT* : 7 bits, ce champ identifie le type du payload (audio, video, image, texte, html, etc.)
- *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 Timestamp* : 32 bits, reflète l'instant où le premier octet du paquet RTP a été échantillonné.
- *Le champ SSRC* : 32 bits, identifie de manière unique la source ; sa valeur est choisie de manière aléatoire par l'application. La liste des CSRC identifie les sources qui ont contribué à l'obtention des données contenues dans le paquet qui contient ces identificateurs. Le nombre d'identificateurs est donné dans le champ CC.
- *Le champ CSRC* : 32 bits, identifie les sources y contribuant.

### 5.8.2. *Le protocole RTCP (Real-time Transport Control Protocol)*

#### 5.8.2.1. L'objet du protocole RTCP

RTCP est un protocole de contrôle associé à RTP, il mesure les performances, par contre, il n'offre pas de garantie. Le protocole RTCP, au-dessus de UDP, est basé sur des transmissions périodiques de paquets de contrôle par tous les participants dans la session. RTCP peut aussi bien utiliser le mode unicast que le mode multicast.

C'est un protocole de contrôle des flux RTP, permettant de véhiculer des informations basiques sur les participants d'une session, et sur la qualité de service.

#### 5.8.2.2. Les fonctions du protocole RTCP

Les fonctions du protocole RTCP sont :

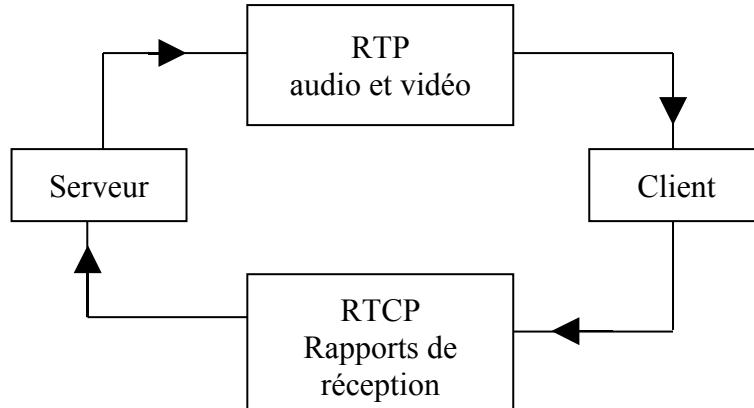
- Un retour de qualité de service lors de la demande de session. Les récepteurs utilisent RTCP pour renvoyer vers la source un rapport sur la qualité de service. Ces rapports comprennent le nombre de paquets perdus, le délai d'aller-retour. Ces informations permettent à la source de s'adapter, c'est-à-dire, par exemple, de modifier le degré de compression pour maintenir la qualité de service.

-Une synchronisation supplémentaire entre médias. Les applications multimédias sont souvent transportées par des flux distincts : par exemple, la voix et l'image, ou même une application numérisée sur plusieurs niveaux hiérarchies, peuvent voir les flux générés suivre des chemins distincts.

-L'identification : les paquets RTCP contiennent des informations d'adresse comme l'adresse d'un message électronique, un numéro de téléphone ou le nom d'un participant à une conférence téléphonique.

-Le contrôle de la session. RTCP permet aux participants d'indiquer leur départ d'une conférence téléphonique ou simplement une identification de leur comportement.

La *Figure 5.08* illustre l'acheminement des paquets RTP et RTCP.



*Figure 5.08* : l'acheminement des paquets RTP et RTCP.

#### 5.8.2.3. Les formats des paquets

RTCP est défini par les paquets suivants :

- sender report (SR) : le rapport envoyé par la (les) source(s).
- receiver report (RR) : le rapport envoyé par le (les) destinataire(s).
- source description (SDES) : la description d'une source.
- goodbye (BYE) : le paquet envoyé quand une source quitte la session et qui contient un champ qui décrit la raison pour laquelle elle quitte la session.

### 5.8.2.3.1. Les paquets RTCP SR :

Les paquets SR sont divisés en trois parties :

- une en-tête de taille fixe qui contient, parmi d'autres informations, la version du protocole, et la longueur du paquet.
- un bloc d'information de la source.
- zéro ou plusieurs blocs de rapport sur les sources qu'il écoute.

Le format du paquet RTCP SR est indiqué sur la *Figure 5.09* suivante:

0	32			
V	P	RRC	Packet type	Length
		SSRC		
		NTP Timestamp		
		NTP Timestamp		
		RTP Timestamp		
		Sender's packet count		
		Sender's octet count		
		SSRC_1		
	Fraction lost		Cumulative number of packets lost	
		Extended highest sequence number received		
		Interarrival jitter		
		LSR		
		DLSR		
		SSRC_2		
		.....		

*Figure 5.09*: le format des paquets RTCP SR.

- *Le champ version* (2 bits)
- *Le champ padding* (1bit) indique qu'il y a du bourrage dont la taille est indiquée dans le dernier octet.
- *Le champ reception report count* (5 bits) : nombre de compte-rendu dans le paquet.
- *Le champ packet type* (8 bits) : 200 pour SR.
- *Le champ length* (16 bits) : longueur du paquet en mots de 32 bits.
- *Le champ SSRC* (32 bits) identification de la source spécifique à l'émetteur.
- *Le champ sender's packet count* (32 bits).
- *Le champ sender's octet count* (32 bits) : les statistiques.
- *Le champ SSRC\_n* (32 bits) numéro de la source dont le flux est analysé.
- *Le champ fraction lost* (8 bits)

- *Le champ cumulative number of packets lost* (24 bits)
- *Le champ extends highest sequence number received* (32 bits)
- *Le champ interarrival jitter* (32 bits). C'est une estimation de l'intervalle de temps d'un paquet de données RTP qui est mesuré avec le timestamp et qui est sous forme d'un entier.
- *Le champ last SR timestamp* (32 bits)
- *Le champ delay since last SR* (32 bits)

*L'information de la source* permet d'informer, pour la source qui envoie ce paquet, le nombre des paquets et d'octets qu'elle a déjà transmis. Elle permet aussi aux destinations d'associer le vrai temps, communiqué dans ce paquet, qui correspond au timestamp de l'échantillon communiqué aussi dans le même paquet.

*Les blocs de rapport* sont le moyen qu'ont les destinations pour communiquer leur vue du réseau en fonction de certains paramètres observés dans la séquence de paquets reçus par cette destination depuis une source donnée. Ces paramètres comprennent le taux de pertes, le moment auquel le dernier SR a été reçu, ainsi que le délai entre le dernier paquet SR reçu et l'envoi du paquet qui contient ce bloc de rapport (un paquet SR ou RR).

#### 5.8.2.3.2. Les paquets RTCP RR :

Les paquets RR sont divisés en deux parties :

- une en-tête de taille fixe qui contient parmi d'autres informations, la version du protocole, et la longueur du paquet ;
- zéro ou plusieurs blocs de rapport sur les sources qu'il écoute.

Le format des paquets RTCP RR est indiqué sur la *Figure 5.10* suivante :

0		32		
V	P	RRC	Packet type	Length
SSRC				
SSRC 1				
Fraction lost		Cumulative number of packets lost		
Extended highest sequence number received				
Interarrival jitter				
LSR				
DLSR				
SSRC 2				
...				

*Figure 5.10* : le format des paquets RTCP RR.

#### 5.8.2.3.3. Les paquets RTCP SDES :

Les paquets de description de la source permettent de communiquer des informations concernant la source qui envoie ce paquet. Information type : le nom de l'utilisateur, son nom, son adresse e-mail, etc.

Le format des paquets RTCP SDES est indiqué sur la *Figure 5.11* suivante :

0	32			
V	P	RRC	Packet type	Length
		SSRC / SSRC 1		
		SDES		
...				

*Figure 5.11*: le format des paquets RTCP SDES.

#### 5.8.3. Le protocole RTSP (*Real Time Streaming Protocol*)

##### 5.8.3.1. Généralités sur le protocole RTSP

La technologie du Streaming utilise un protocole en temps réel, le RTSP. Ce protocole, suivant le RFC 2326, est un protocole de contrôle pour les présentations multimédias au niveau du modèle client-serveur, il est conçu pour adresser les besoins pour le Streaming multimédia sur les réseaux IP.

Ce protocole a été présenté à l'IETF en Octobre 1996 par Real Networks et Netscape Communications Corporation avec le soutien de plus de 40 entreprises actives dans les médias, dont Intel. Le standard fût finalement publié en Avril 1998 par l'IETF. RTSP est un protocole de présentation multimédia client/serveur, destiné à satisfaire les critères d'efficacité d'acheminement des données médias streamés sur des réseaux IP.

Il s'agit d'un protocole du niveau 7 (applications) dont la particularité est de *pouvoir afficher un flot de données sans devoir attendre la fin de l'envoi*. De ce fait, il est fortement prédisposé à la transmission des fichiers multimédias tel que le son, les extraits de vidéo enregistré, ou des images filmées en temps réel. Il se sert pour cela des protocoles RTP, RTCP, et RSVP. Si H.323 a été conçu pour la vidéoconférence (pour un nombre modéré de

personnes), RTSP lui est conçu pour diffuser efficacement des données audiovisuelles à un grand nombre de personnes.

#### 5.8.3.2. Les principes du protocole RTSP

RTSP a pour cible principale les objets média distants continus (un film par exemple), dont le transfert et le traitement de données doivent être effectués en temps réel. Ainsi, RTSP

a été développé en tant que protocole « temps réel » et conçu pour transférer et contrôler différents flux simultanés et synchronisés d'objets médias continus, par exemple un film et sa bande son.

Le protocole RTSP se base sur un modèle « client/serveur », c'est-à-dire qu'il permet de contrôler à distance le déroulement des opérations. Ainsi, il est possible, lors de la visualisation d'un film préenregistré, d'envoyer des ordres tel que « Démarrer ; Stop ». Ces commandes sont envoyées par le client via le protocole RTSP sous forme de requêtes d'accès et sont exécutées par le serveur.

RTSP ne crée pas une connexion au sens TCP mais une session, maintenue par le serveur, identifiée par un numéro. La session est la transaction établie pour visionner le flux.

Ici, la notion de session est importante. En effet, le protocole RTSP ne se contente pas de transmettre les requêtes du client et les données du serveur, mais il est capable de reconnaître à quel instant un paquet de données doit être transmis au client, ceci au moyen d'une «signature temporelle» inscrite sur chaque paquet. Ainsi, si le serveur se rend compte qu'un paquet est en retard, il ne l'envoie pas au client, ce qui garantit à ce dernier une fluidité accrue puisqu'il n'est pas obligé de traiter inutilement ces données.

Durant la session, le client envoie d'abord au serveur un message SETUP pour initialiser la transaction. Il y a négociation des ports, du mode de diffusion (unicast, multicast), et la vitesse de connexion. Dès lors, en envoyant une requête PLAY, le flux est diffusé. Il est possible de faire une pause pendant la lecture, au moyen de la requête PAUSE, et de l'arrêter complètement, c'est-à-dire mettre fin à la session, en émettant la requête TEARDOWN.

Une dernière caractéristique du protocole RTSP est que le client émet régulièrement un feed-back au travers du protocole RTCP au serveur, sous la forme d'un rapport qualitatif et

quantitatif de la transmission. Ainsi, le serveur peut adapter son flux de données destiné au client.

#### 5.8.3.3. Les méthodes du protocole RTSP

Le *Tableau 5.02* montre quelques méthodes utilisées par le protocole RTSP.

Ces méthodes sont invoquées, soit par le client vers le serveur ( $C \rightarrow S$ ), soit une réponse du serveur ( $S \rightarrow C$ ) vers le client.

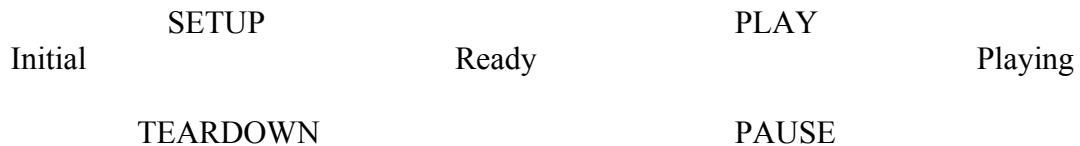
Méthodes	Direction
<b>DESCRIBE</b>	$C \rightarrow S$
<b>GET_PARAMETER</b>	$C \rightarrow S, S \rightarrow C$
<b>OPTIONS</b>	$C \rightarrow S, S \rightarrow C$
<b>PAUSE</b>	$C \rightarrow S$
<b>PING</b>	$C \rightarrow S, S \rightarrow C$
<b>PLAY</b>	$C \rightarrow S$
<b>REDIRECT</b>	$S \rightarrow C$
<b>SETUP</b>	$C \rightarrow S$
<b>SET_PARAMETER</b>	$C \rightarrow S, S \rightarrow C$
<b>TEARDOWN</b>	$C \rightarrow S$

*Tableau 5.02* : les méthodes RTSP.

Parmi les méthodes ci-dessus, seulement 5 contribuent à définir l'état de la session :

- SETUP : Le client demande au serveur l'allocation des ressources pour un flux et commence une session RTSP ;
- PLAY : Le client demande la transmission des données d'un flux alloué selon SETUP ;
- RECORD : Le client initie l'enregistrement selon les paramètres de la description de la présentation ;
- PAUSE : Le client arrête temporairement le flux sans libérer les ressources du serveur.
- TEARDOWN : Le client demande de libérer les ressources associées au flux. La session RTSP cesse d'exister sur le serveur.

La *Figure 5.12* illustre les différents états de transition des méthodes RTSP.



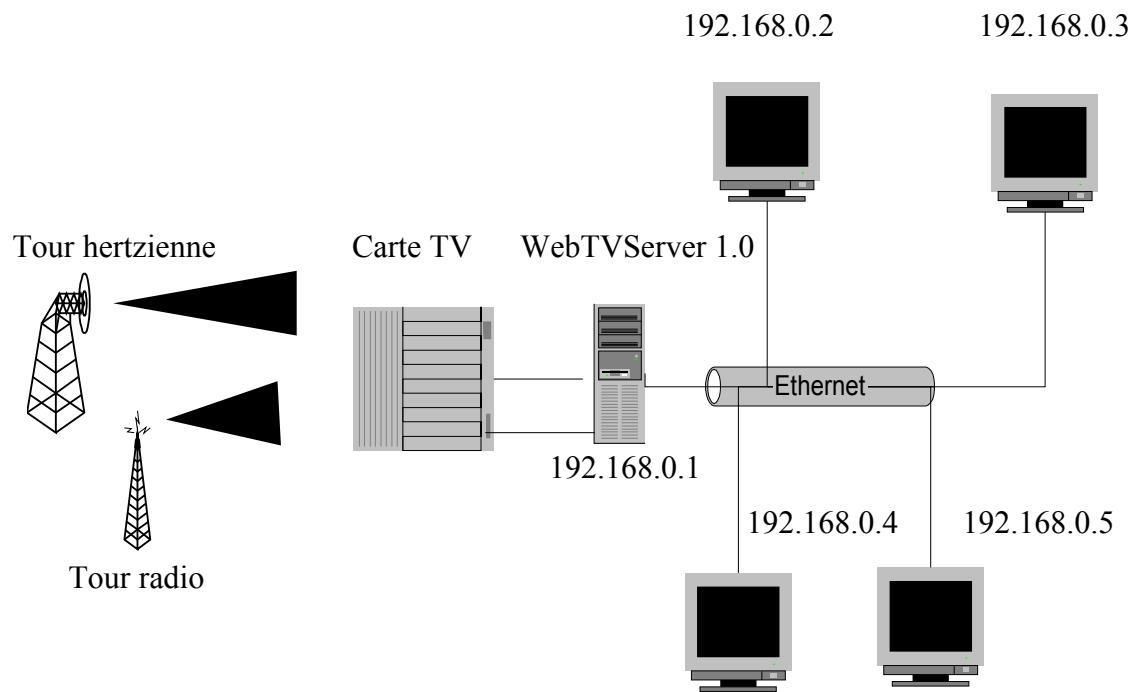
*Figure 5.12 : les états de transition des méthodes RTSP.*

## CHAPITRE 6 : SIMULATION DE LA MISE EN ŒUVRE D'UNE WEB TV

### 6.1 Mise en place d'un système de WEB TV

Dans ce chapitre, on étudiera le fonctionnement d'une WEB TV sur un réseau local Ethernet 100Mbps.

La *Figure 6.01* illustre l'architecture sur laquelle notre simulation va se baser :



*Figure 6.01 : l'architecture du réseau local pour la simulation de la WEB TV*

### **6.1.1. Mise en place du réseau local**

#### **6.1.1.1. Installation matérielle**

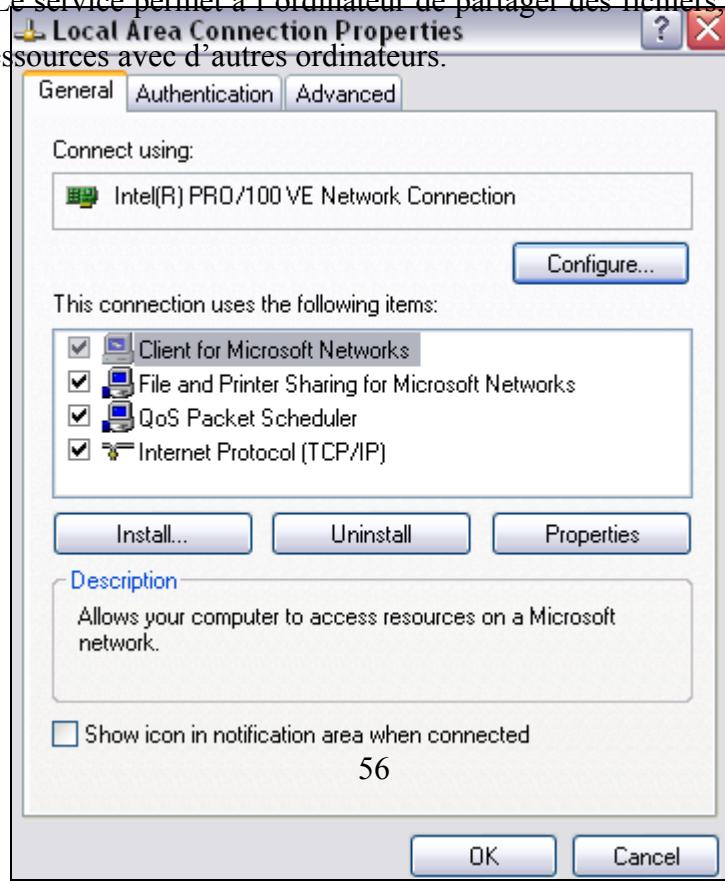
Pour pouvoir communiquer entre eux, chaque ordinateur doit posséder une carte réseau.

Pour installer la carte réseau, il faut éteindre l'ordinateur, ouvrir le boîtier et insérer la carte réseau dans un port PCI libre, puis redémarrer l'ordinateur. La carte est automatiquement détectée par l'ordinateur et il fait une mise à jour des pilotes disponibles. Eventuellement, il demande d'insérer la disquette ou le CD-ROM contenant les pilotes.

#### **6.1.1.2. Installation logicielle**

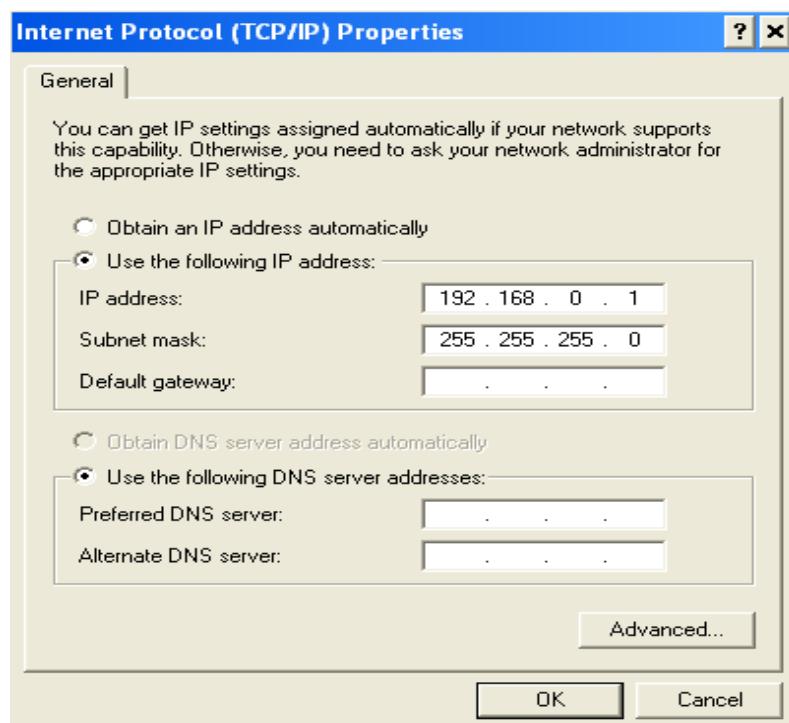
L'installation du réseau nécessite l'installation des composants réseaux suivants, comme indiquée sur la *Figure 6.02*:

- **Carte** : c'est le périphérique matériel qui permet à un ordinateur de se connecter à d'autres.
- **Client** : Le client est un composant logiciel qui permet de connecter un ordinateur à d'autres.
- **Protocole** : Un protocole est une langue parlée par l'ordinateur. Deux ordinateurs doivent utiliser le même protocole pour communiquer.
- **Service** : Le service permet à l'ordinateur de partager des fichiers, des imprimantes et d'autres ressources avec d'autres ordinateurs.



*Figure 6.02 : les composants nécessaires*

Pour assigner une Adresse IP, il faut cliquer sur *Properties* de *Internet Ptotocol (TCP/IP)*, représenté sur la *Figure 6.03* :



*Figure 6.03 : la configuration de l'adresse IP*

#### 6.1.1.3. Le câblage

Le câblage utilisé pour ce réseau est dit « droit » : il s’agit de l’interconnexion d’une machine avec un *switch*, un *hub*, ou un *concentrateur local*. Les câbles ont, à leur extrémité, des prises RJ45. Ces derniers sont insérés dans un port libre de l’équipement d’interconnexion.

### **6.1.2. Installation de la carte TV**

Afin de pouvoir capter des émissions de Télévision, le serveur doit être pourvu d'une Carte TV. Dans notre simulation, nous avons utilisé une carte TV dont la référence est PV-BT878P+ w/FM.

#### **6.1.2.1. Installation matérielle**

On insère la carte TV dans un port PCI libre. Après cela, le système d'exploitation détecte automatiquement sa présence au démarrage de l'ordinateur.

#### **6.1.2.2. Installation logicielle**

On installe alors le pilote fourni avec la carte, et qui correspond au système d'exploitation. Plusieurs versions existent pour Windows 98, XP, 2000, etc. Nous avons installé le pilote pour Windows XP.

Lorsque notre carte TV est installée correctement, on a une interface comme indiquée sur la *Figure 6.04*:



*Figure 6.04 : interface de contrôle TV / FM.*

Les boutons FM et TV servent à basculer le mode.

Pour son utilisation, se référer au *Manuel d'utilisation* ou le CD fourni avec la carte TV. C'est cette interface qui va nous servir de pilote pour contrôler la sortie de la carte TV.

### **6.1.3. Mise en place du Serveur Web**

Nous avons mis en place un serveur Web afin de servir de système d'information pour le client qui souhaite s'informer des programmes actuellement diffusés en ligne ou des fichiers médias disponibles en Streaming. Nous avons utilisé Tomcat5.0 de Apache Foundation.

## 6.2 Le langage JAVA et le Streaming

### 6.2.1. Présentation de l'API JMF

L'objectif de ce paragraphe est de présenter une manière d'effectuer du Streaming en utilisant le langage Java. Il présentera essentiellement l'API JMF (Java Media Framework) et ses différentes fonctionnalités.

### 6.2.2. Une brève description de l'API JMF

Il faut dans un premier temps installer une JDK (Java Development Kit) afin de pouvoir développer des programmes en Java. On peut ensuite installer JMF qui fournit les APIs nécessaires pour effectuer du Streaming.

JMF est donc une API permettant l'exploitation du Streaming avec le langage java. Elle offre de nombreuses possibilités et est très simple d'utilisation. Elle a été conçue dans sa version 2 pour répondre aux attentes suivantes :

- Permettre la capture de données multimédias
- Permettre le développement d'application java utilisant le streaming ou les conférences vidéos.
- Permettre l'accès à un large type de données
- Offrir un support pour le protocole RTP (Real-Time Transport Protocol)

### 6.2.3. Décomposition de l'API JMF

JMF se décompose en deux modules distincts :

- L'API de base : Elle fournit toute une architecture permettant de gérer l'acquisition, le traitement et l'affichage de données multimédias. On peut alors facilement à, l'aide de JMF, créer une applet ou une application qui présente capture, manipule ou enregistre des flux multimédia. On trouve alors différents outils comme les Players qui vont permettre la visualisation et le traitement des données. On pourra alors grâce à eux traiter le flux vidéo et permettre les options que l'on souhaite sur le lecteur media.
- L'API RTP : Jusque là, JMF ne permettait que de lire, traiter et présenter un flux arrivant à un utilisateur. Grâce à l'API RTP on va maintenant pouvoir transmettre un flux et ainsi créer son propre serveur de streaming. On peut maintenant capturer un flux à partir d'une caméra ou un micro et le transmettre à différents utilisateurs ou encore centraliser un ensemble vidéo et sons et les transmettre sur demande.

Enfin JMF est prévue pour être étendue. En effet elle permet de développer ses propres plugins afin d'effectuer des traitements particuliers de fichiers audios ou vidéos ou encore de traiter certains formats non supportés pour des besoins particuliers.

#### 6.2.3.1. La Gestion des données

L'API de base contient les outils nécessaires à la manipulation de données multimédia. Elle fournit les méthodes pour gérer des données média (DataSource), les lire (Player) ou encore effectuer des traitements (Processor). Les lecteurs JMF utilisent en général des "Datasource"(définis par l'API JMF) pour gérer le transfert du contenu d'un media. Il encapsule un ensemble de données sur le transfert à effectuer :

- L'emplacement du fichier que l'on veut transporter
- Le protocole que l'on va utiliser pour délivrer le flux
- Les outils utilisés pour le transfert

Un DataSource est identifié par un objet MediaLocator ou URL permettant de spécifier l'emplacement du media à gérer. Un MediaLocator est similaire à l'objet URL et peut être construit à partir de celui-ci. La seule différence est qu'il ne nécessite pas que le handler(gestionnaire) du protocole utilisé soit installé sur la machine pour être utilisé.  
ex : ftp://... On pourrait l'utiliser même si le handler ftp n'était pas installé

Il existe alors deux types de DataSources en fonction de l'émetteur et du récepteur :

- Pull Data-Source : Le client initialise le transfert de données et contrôle le flot de données avec le PullDataSource
- Push Data-Source : Le serveur initialise le transfert de données et contrôle le flot de données avec le PushDataSource

Les formats de fichiers multimédias sont gérés par des classes de type Format contenant les attributs spécifiques du format utilisé.

#### 6.2.3.2. Les Managers

Pour simplifier l'utilisation des différentes fonctions de l'API, JMF fournit des classes Managers permettant d'accéder à tous les services principaux. Cela permet de centraliser les différentes fonctions :

- Manager : Cette classe permet la construction des objets principaux, autrement dit les Players, Processors, DataSources et DataSinks(permettre l'enregistrement d'un flux ou la capture)
- PackageManager : Maintient un registre de packages contenant des classes JMF personnalisées (Players particuliers ...)
- CaptureDeviceManager : Maintient une liste des périphériques de capture valable dans JMF
- PlugInManager : maintient une liste des plugins JMF de traitement utilisables (Multiplexers, Demultiplexers, Codecs, Effects, Renderers).

#### 6.2.3.3. Les objets Players

Un objet Player prend un flux en entrée sous la forme d'un DataSource et le présente (effectue un rendu) à un temps donné. On peut alors le configurer pour obtenir le rendu que l'on désire. Il est en fait le lecteur qui va nous permettre d'interpréter le flux et l'afficher. Il peut se trouver dans six états différents en respectant l'ordre suivant :

- Unrealized : Le Player vient d'être instancié mais ne connaît rien sur le média qu'il doit traiter.
- Realizing : A l'appel de la méthode "realize" il passe dans cet état. Cela consiste à acquérir les ressources qu'il n'a besoin d'acquérir qu'une fois. Cela comprend le rendu et les différentes informations dont il a besoin sur le media à présenter
- Realized : Le player passe dans cet état à la sortie de la méthode "realize". Il connaît toutes les ressources dont il a besoin, le type du media à présenter et la façon dont il doit le faire.
- Prefetching : Il passe dans cet état à l'appel de "prefetch". Le player précharge le média, prend un accès exclusif sur les ressources dont il a besoin et prépare tout ce dont il a besoin pour lancer la lecture.
- Prefetched : Le player est près à être lancé.
- Started : Il a commencé la lecture ou la commencera à un temps donné.

Le Player envoie des "TransitionEvents" lorsqu'il passe d'un état à un autre.

#### 6.2.3.4. L'objet Processor

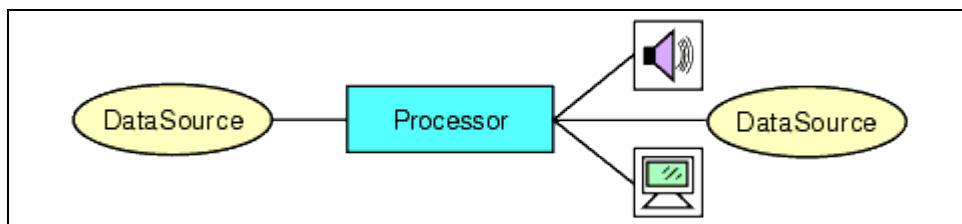
Les données d'un flux média sont soumises à un traitement avant d'être présentées à un utilisateur. On pourra alors effectuer les opérations suivantes avant la présentation du média :

- Si le flux est multiplexé, on peut extraire les pistes individuelles
- Si les pistes sont compressées, elles sont décodées et décompressées ;

On peut convertir le flux dans un autre format ; On peut appliquer des filtres d'effets sur le flux.

L'objet Processor est en fait un Player particulier pouvant contrôler les traitements effectués sur le flux. Il peut présenter un média ou proposer un nouveau DataSource en sortie.

L'objet Processor est illustré sur la *Figure 6.05*.



*Figure 6.05* : l'objet Processor

#### **6.2.4. Le fonctionnement de l'API JMF : l'API RTP**

L'API va nous permettre de gérer des sessions RTP très simplement. Tout d'abord comme pour l'API de base JMF, elle fournit une classe Manager (RTPManager) qui centralise toutes les fonctions liées à une session RTP. Cette classe va alors nous permettre l'envoi et la réception de flux RTP.

##### **6.2.4.1. Réception d'un flux RTP**

L'accès à une ressource par un serveur RTP se fait en utilisant une adresse spécifique de la forme `rtp://[adresse ip]:[port]/[type]/`. On peut alors créer un objet MediaLocator à partir de cette adresse qui sera utilisé comme source du flux RTP. On peut ensuite l'API de base RTP pour présenter le flux arrivant dans une application ou une applet.

##### **6.2.4.2. Emission d'un flux RTP**

Dans un premier temps l'objet Processor va nous permettre d'appliquer les traitements nécessaires avant la transmission sur le réseau. On va alors pouvoir paramétriser le type d'encodage, la qualité et tout les paramètres liés au fichier que l'on souhaite transporter. Une fois ces traitements effectués sur le fichier on utilise le PushDatasource qu'il nous renvoie en

sortie pour créer les flux d'envoie. Le RTPManager nous permet alors de créer une session RTP pour chaque flux que l'on souhaite transiter.

### ***6.2.5. Développement des Applications Clients et Serveurs***

#### **6.2.5.1. WebTVServer 1.0**

WebTVServer 1.0 est une solution logicielle de diffusion vidéo, que nous avons développé dans le cadre de ce mémoire.

Nous avons conçu WebTVServer 1.0 pour diffuser des vidéos, de l'audio et des émissions télévisées obtenues à partir d'une carte TV sur des réseaux haut débit..

WebTVServer 1.0 a été programmé pour reconnaître les formats MP3 et MPEG, AVI, MOV et WAV.

##### ***6.2.5.1.1. L'objectif de WebTVServer 1.0***

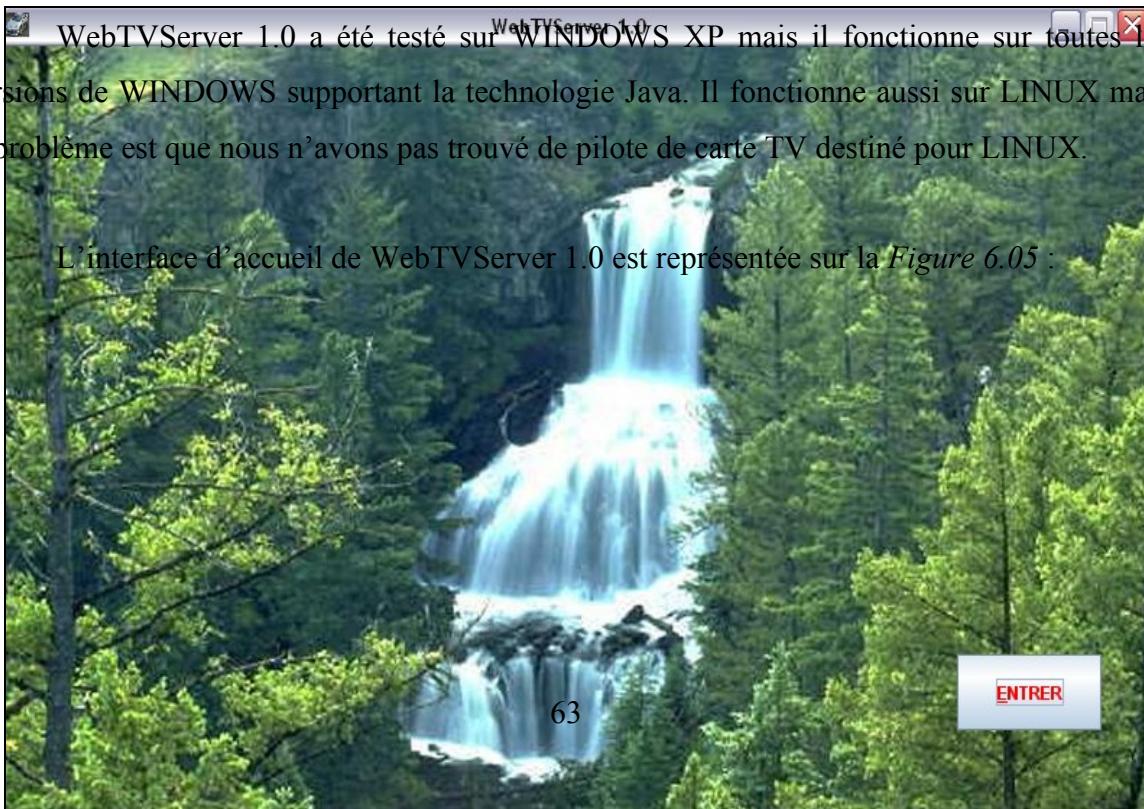
Son objectif primordial est de transmettre les émissions télévisées vers une machine distante à travers un réseau IP : il peut utiliser le mode Unicast, ou le mode Multicast (vers un groupe dynamique de machines que les clients rejoignent ou quittent).

##### ***6.2.5.1.2. Configuration requise***

Une machine Pentium III 733 Mhz munie de 128 Mo de mémoire devrait suffire pour diffuser un flux sur un réseau mais étant donné qu'il s'agit quand même d'un serveur, une configuration supérieure serait d'autant plus bénéfique.

WebTVServer 1.0 a été testé sur WINDOWS XP mais il fonctionne sur toutes les versions de WINDOWS supportant la technologie Java. Il fonctionne aussi sur LINUX mais le problème est que nous n'avons pas trouvé de pilote de carte TV destiné pour LINUX.

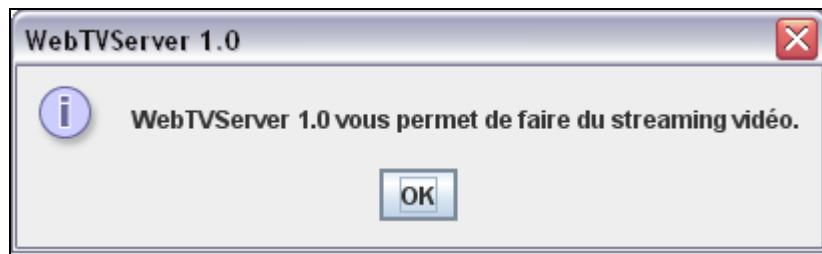
L'interface d'accueil de WebTVServer 1.0 est représentée sur la *Figure 6.05* :



*Figure 6.05 : l'interface d'accueil de WebTVServer 1.0*

#### *6.2.5.1.3. L'utilisation de WebTVServer 1.0*

Après avoir cliqué sur le bouton *ENTRER*, un petit message d'information apparaît :



*Figure 6.06 : message d'information de WebTVServer 1.0*

Ensuite, il demande l'*ADRESSE IP* vers lequel diriger le flux.



*Figure 6.07 : adresse IP de destination*

Mettons 192.168.0.2. C'est l'adresse IP d'un client.

Puis le second paramètre demandé est le numéro de port :

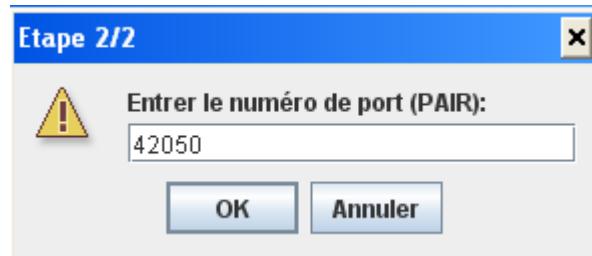


Figure 6.08 : port de destination du flux

Pour cela, se référer à la liste des ports dans l' ANNEXE 2 : LES PORTS pour éviter un conflit interne entre programmes.

**Remarque:** Le numéro de port doit être un numéro PAIR. En effet, lorsque le client contacte le serveur WebTVServer 1.0 ou tout autre serveur Streaming en cas réel sur Internet, les communications RTP se font sur le numéro pair, et les contrôles RTCP se font sur le numéro pair immédiatement supérieur.

Le *Tableau 6.01* donne la liste des ports couramment utilisés dans ce type de transmission:

VIDEO	AUDIO
22222	22224
2222	2224
42050	42506
42050	42052

Tableau 6.01 : tableau des ports couramment utilisés

On peut maintenant choisir la source à transmettre :



Figure 6.09 : le choix de la source à transmettre.

#### 6.2.5.2. WebTVPlayer 1.0

WebTVPlayer 1.0, voir *Figure 6.10*, est une solution logicielle de lecture vidéo, que nous avons personnellement développé dans le cadre de ce mémoire.

Nous avons conçu WebTVPlayer 1.0 pour la lecture des vidéos, de l'audio et des émissions télévisées diffusées en direct sur un réseau IP.

WebTVPlayer 1.0 a été programmé pour reconnaître les formats MP3 et MPEG, AVI, .MOV et WAV.

#### 6.2.5.2.1. L'objectif de WebTVPlayer 1.0

Comme son nom l'indique, il a été surtout développé afin de visualiser, en temps réel, des émissions TV diffusées à partir d'un serveur distant, notamment WebTVServer 1.0.

Toutefois, il ne se limite pas à cette tâche. En effet, il peut servir de lecteur multimédia comme le fait Windows Media Player, mais à la différence que WebTVPlayer 1.0 n'attend pas que le fichier soit chargé entièrement pour le jouer. Il utilise la technique du Streaming pour cela.

Son autre fonctionnalité intéressante est qu'il peut lire, non seulement les fichiers qui se trouvent sur votre disque, mais aussi les fichiers multimédia qui se trouvent sur un serveur WEB distant.



Figure 6.10: l'accueil de WebTVPlayer 1.0

#### 6.2.5.2.2. L'utilisation de WebTVPlayer 1.0

L'utilisation de WebTVPlayer 1.0 a été expressément rendue facile pour l'utilisateur. Il ne comporte certes que trois options, mais ce sont les options les plus essentielles.

- Un choix sur le bouton LOCALE ouvre une fenêtre qui parcourt le disque dur local, voir *Figure 6.11*:

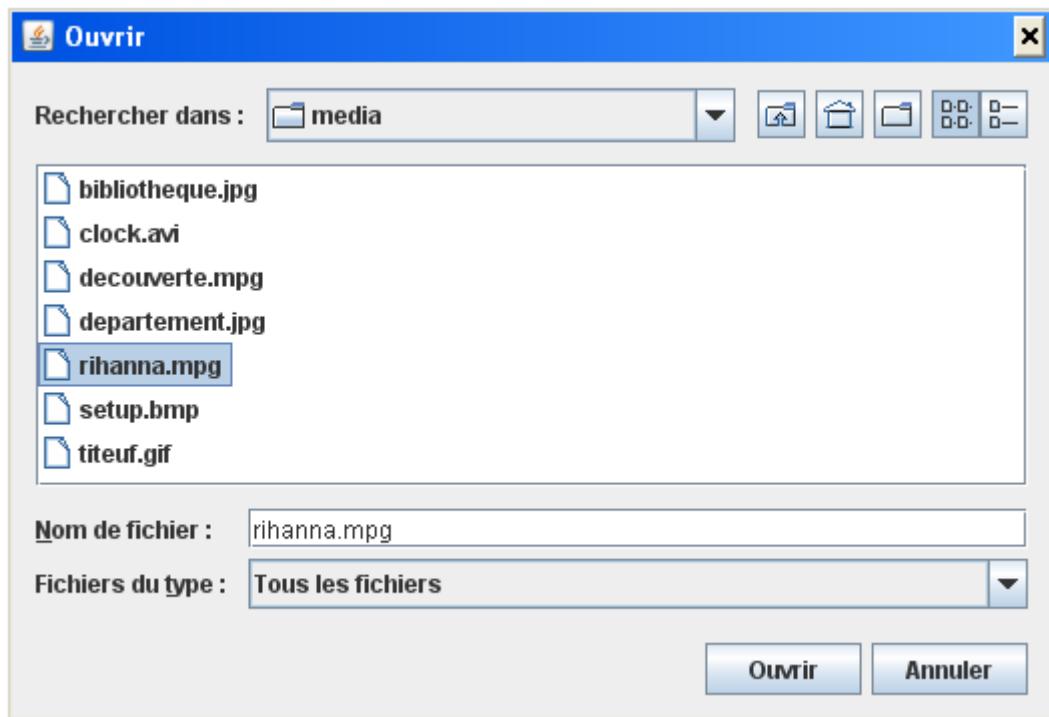


Figure 6.11 : lecture d'un fichier local

- Le bouton **TV en Direct** permet de se connecter au serveur Streaming qui diffuse les émissions TV. Le client peut alors visualiser, en direct, une émission télévisée transmise sur le réseau.

Un délai de 16 secondes a été défini pour attendre une réponse du serveur si celui-ci transmet des flux RTP dans l'instant en cours. Sinon, le client WebTVPlayer 1.0 arrête le processus d'attente et il quitte le programme.

- Le bouton  permet de taper le nom d'un fichier situé sur le serveur WEB.

Exemple :

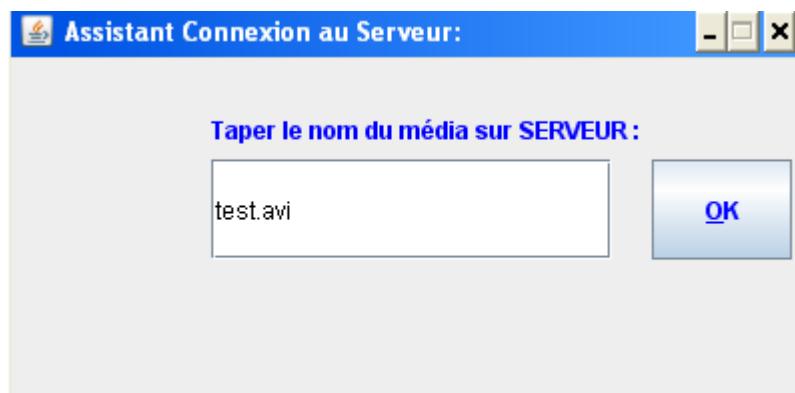


Figure 6.12 : une source média sur un serveur WEB.

#### 6.2.5.2.3. Limitations de WebTVPlayer 1.0

WebTVPlayer 1.0 connaît quelques limitations :

- Il ne lit pas les fichiers au format .WMV, WMA et .DAT ; ces formats étant propriétaires, les codecs correspondants sont assez difficiles à trouver.
- WebTVPlayer 1.0 soulève quelques difficultés sur les réseaux équipés de firewalls.

#### 6.2.5.3. Le fichier de configuration « config.jsa »

L'utilisation de WebTVPlayer 1.0 a été expressément rendue aussi facile que possible pour l'utilisateur moyen et les mécanismes qui s'opèrent durant les sessions entre client et serveur sont totalement transparents pour l'utilisateur.

Un fichier de configuration, appelé ***configure.jsa*** est cependant mis à la disposition de l'utilisateur avancé qui désire faire appliquer ses paramètres de connexion personnels.

Voici la structure du fichier ***configure.jsa*** :

```
<Adresse IP [du groupe vers lequel s'abonner en mode Multicast] de l'envoi de flux>
<Numéro de port pour le service Web>
<Numéro de port pour la session RTP vidéo>
< Numéro de port pour la session RTP audio>
```

Exemple :

```
192.168.0.1
80
42050
42506
```

Dans le cas où le client souhaite lire en mode Streaming un fichier multimédia se trouvant sur un serveur WEB distant, l'adresse IP de ce serveur WEB est 192.168.0.1 et le numéro de port par lequel WebTVPlayer 1.0 extrait le flux est le numéro de port 80, numéro par défaut du service WEB sur la plupart de tous les serveurs WEB.

Les ports 42050 et 42506 ne sont pas utilisés dans ce cas.

Dans le cas où l'utilisateur désire visionner des émissions de télévision en temps réel et en direct sur un serveur qui offre ce service, le numéro de port de la session RTP pour la vidéo sera le numéro 42050 et celui de l'audio sera 42506.

Rappelons que nous avons vu dans la partie théorique que si le contrôle de flux est assuré par le protocole RTCP ; ce dernier est désigné pour travailler conjointement avec RTP. Chacun d'eux utilise un port séparé d'une paire de ports. RTP utilise le port pair, et RTCP le port impair immédiatement supérieur. Il est donc à éviter de configurer le numéro de port par un numéro de port impair.

En outre, l'adresse IP 192.168.0.1 identifie une machine en particulier et l'utilisateur peut visionner ses émissions TV dans le cas où :

- Le serveur transmet en mode Unicast et 192.168.0.1 est l'adresse IP où il envoie le flux, c'est donc l'adresse IP de l'utilisateur ;
- Le serveur émet en mode Multicast à l'adresse 192.168.0.255 ; l'utilisateur fait alors partie des ordinateurs du sous-réseau.

## CONCLUSION

La mise en place d'un système de WEB TV destiné à la diffusion de contenus multimédia s'inscrit dans le cadre de l'utilisation des réseaux comme support de nouveaux services.

Hier encore inimaginable, la Télévision sur Internet est aujourd'hui possible grâce aux progrès effectués dans le domaine des réseaux à haut débit, tant au niveau des infrastructures d'interconnexion qu'au niveau de la boucle locale.

La convergence de l'Internet et de l'Audiovisuel en 1990 a entraîné l'essor des techniques numériques. Plusieurs normes de compression ont été développées.

Ce mémoire nous a permis d'explorer les capacités et les limites de la transmission en temps réel sur un réseau IP.

Désormais, on sait que tous les types de média peuvent être transmis sur un réseau IP, et ce, en temps réel, moyennant des infrastructures réseaux pouvant offrir des débits assez élevés.

Ceci est d'un apport très considérable car un portail s'ouvre entre les médias existants et le réseau de transit IP.

Le développement d'un projet de Télévision sur Internet nécessite la maîtrise d'une grande série de disciplines, entre autres la connaissance des signaux de télévision, le traitement numérique des signaux, les réseaux, la programmation et l'implémentation des protocoles temps réels .

La combinaison de ces derniers nous a permis, non seulement de mener à terme notre travail de mémoire, mais également nous a ouvert la voie vers d'autres perspectives d'avenir, à savoir :

- l'étude des nouvelles normes de compression proposées par les organisations de standardisation, pour une qualité télévisuelle satisfaisante ;
- le développement d'interface WEB directement disponible pour le client ;

Cela est laissé entre les mains des étudiants futurs qui souhaiteront poursuivre les enjeux de recherche de ce mémoire.

## **ANNEXE 1 : LES STANDARDS MONDIAUX DE TELEVISION**

Les deux grands standards de base (CCIR et FCC) regroupent seulement les standards qui ont le même nombre de lignes d'images et le même nombre d'images affichées par seconde. Mais il existe aussi d'autres paramètres nécessaires à définir pour une émission TV, c'est ce qui explique l'existence de huit (8) standards dans le monde. Ils définissent les caractères des signaux audio et vidéo et les caractéristiques de leur émission.

Les caractéristiques de ces divers standards sont données *ANNEXE 1.01*.



Standard		B/G CCIR	D/K	H	I	K1 (K')	L	M FCC	N
Caractéristiques des signaux	Affectation	VHF / UHF	VHF / UHF	UHF	VHF / UHF	VHF / UHF	VHF / UHF	VHF / UHF	VHF / UHF
	Nombre de lignes d'une image	625	625	625	625	625	625	525	625
	Fréquence trame (Hz)	50	50	50	50	50	50	60	50
	Fréquence ligne (Hz)	15 625	15 625	15 625	15 625	15 625	15 625	15 750	15 625
	Durée impulsion de synchronisation ligne( $\mu$ s)	4,7	4,7	4,7	4,7	4,7	4,7	5	5
	Durée impulsion de suppression ligne ( $\mu$ s)	12	12	12	12	12	12	11	10,9
	Durée de suppression trames (lignes)	25	25	25	25	25	25	19 à 21	19 à 25
	Largeur de bande vidéo (Mhz)	5	6	5	5,5	6	6	4,2	4,2
Caractéristiques d'émission	Largeur du canal (Mhz)	7 (B) /8 (G)	8	8	8	8	8	6	6
	Espacement entre Porteuses IMAGE – SON (Mhz)	+5,5	+6,5	+5,5	+6	+6,5	+6,5	+4,5	+4,5
	Largeur de la Bande latérale résiduelle (Mhz)	0,75	0,75	1,25	1,25	1,25	1,25	0,75	0,75
	Type de modulation IMAGE	C3F négative	C3F Négative	C3F Négative	C3F Négative	C3F Négative	C3F positive	C3F Négative	C3F Négative
	Type de modulation SON	F3E	F3E	F3E	F3E	F3E	A3E	F3E	F3E
	Déviation de fréquences (SON) (Khz)	$\pm 50$	$\pm 50$	$\pm 50$	$\pm 50$	$\pm 50$	-	$\pm 25$	$\pm 25$
	Puissance vidéo /son	10 : 1 ou 20 : 1	5 : 1 ou 10 : 1	5 : 1 ou 10 : 1	5 : 1	10 : 1	10 : 1	5 : 1 10 : 1	5 : 1 10 : 1

Annexe 1.01 : Les différents standards mondiaux de TV

## ANNEXE 2 : LES PORTS

Liste des ports couramment utilisés pour obtenir des renseignements sur des systèmes informatiques ou un accès à ces derniers :

<b>Service ou application</b>	<b>Port / Protocole</b>
echo	7/tcp
systat	11/tcp
chargen	19/tcp
ftp-data	21/tcp
ssh	22/tcp
telnet	23/tcp
smtp	25/tcp
nameserver	42/tcp
whois	43/tcp
tacacs	49/udp
Dns-lookup	53/udp
Dns-zone	53/tcp
Oracle-sqlnet	66/tcp
Tftp	69/udp
Finger	79/tcp
http	80/tcp
Port web de recharge	81/tcp
Kerberos ou port web de recharge	88/tcp
Pop2	109/tcp
Pop3	110/tcp
Sunrpc	111/tcp
Sqldrv	118/tcp
nntp	119/tcp
Ntrcp-or-dce	135/tcp
Netbios	139/tcp
Imap	143/tcp
Snmp	161/udp
Snmp-trap	162/udp
Bgp	179/tcp
Snmp-checkpoint	256/tcp
Idap	389/tcp
Netware-ip	396/tcp
Timnuktu	407/tcp
https/ssl	443/tcp
Ms-smb-alternat	445/tcp/udp
Ipsec-internet-key-exchange (ike)	500/udp
rlogin	513/tcp
Rwho	513/udp
Rshell	514/tcp

Syslog	514/udp
Printer	515/tcp/udp
Router	520/tcp
Netware-ncp	524/tcp
Remotelypossible	799/tcp
Socks	1080/tcp
Bmc-patrol-db	1313/tcp
Notes	1352/tcp
Ms-sql	1433/tcp
Citrix	1494/tcp
Sybase-sql-anywhere	1498/tcp
Ingress-lock	1524/tcp
Oracle-srv	1525/tcp
Oracle-tli	1527/tcp
Pptp	1723/tcp
Winsock-proxy	1745/tcp
Remotely-proxy	2000/tcp
Cisco-mgmt	2001/tcp
Nfs	2049/tcp
Compaq-web	2301/tcp
Openview	2447/tcp
Realsecure	2998/tcp
Ms-active-dir-global-catalog	3268/tcp
Bmc-patrol-agent	3300/tcp
Mysql	3306/tcp
Ssql	3351/tcp
Ms-termServ	3389/tcp
Cisco-mgmt	4001/tcp
Nfs-locked	4045/tcp
Pcanywhere	5631/tcp
Vnc	5800/tcp
Xwindows	6000/tcp
Cisco-mgmt	6001/tcp
Apc	6549/tcp
Irc	6667/tcp
Web	8000/tcp
Web	8001/tcp
Web	8002/tcp
Web	8080/tcp
Cisco-remote	9001/tcp
Netbus	12345/tcp
Quake	26000/tcp
Backorifice	31337/udp
Rcp-solaris	32771/tcp
Snmp-solaris	32780/udp
Reachout	43188/tcp
pcanywhere	65301/tcp

### **ANNEXE 3 : LES DIALOGUES RTSP**

Voici un exemple de dialogue :

Les requêtes du Client sont notées par C: et les réponses du Serveur par S: .

C: SETUP movie.Mjpeg RTSP/1.0

C: CSeq: 1

C: Transport: RTP/UDP; client\_port= 25000

S: RTSP/1.0 200 OK

S: CSeq: 1

S: Session: 123456

C: PLAY movie.Mjpeg RTSP/1.0

C: CSeq: 2

C: Session: 123456

S: RTSP/1.0 200 OK

S: CSeq: 2

S: Session: 123456

C: PAUSE movie.Mjpeg RTSP/1.0

C: CSeq: 3

C: Session: 123456

S: RTSP/1.0 200 OK

S: CSeq: 3

S: Session: 123456

C: PLAY movie.Mjpeg RTSP/1.0

C: CSeq: 4

C: Session: 123456

S: RTSP/1.0 200 OK

S: CSeq: 4

S: Session: 123456

C: TEARDOWN movie.Mjpeg RTSP/1.0

C: CSeq: 5

C: Session: 12345

S: RTSP/1.0 200 OK

S: CSeq: 5

S: Session: 123456

## ANNEXE 4 : QUELQUES PROFILS SOURCES JAVA

### La Réception de flux audio et vidéo en utilisant le protocole RTP :

Le principe est le suivant :

- Ouvrir une session RTP,
- Ecoute du *NewReceiveStreamEvent* à partir de *ReceiveStreamListener*,
- Creation d'un Player.

```
import java.io.*;
import java.awt.*;
import java.net.*;
import java.awt.event.*;
import java.util.Vector;
import javax.media.*;
import javax.media.rtp.*;
import javax.media.rtp.event.*;
import javax.media.rtp.rtcp.*;
import javax.media.protocol.*;
import javax.media.protocol.DataSource;
import javax.media.format.AudioFormat;
import javax.media.format.VideoFormat;
import javax.media.Format;
import javax.media.format.FormatChangeEvent;
import javax.media.control.BufferControl;
public class AVReceive2 implements ReceiveStreamListener, SessionListener,
    ControllerListener
{
    public AVReceive2(String sessions[]) {
        //Constructeur
        .....
    }
    protected boolean initialize() {
        try {
```

```

        // Ouvrir la session RTP
    }

    catch (IllegalArgumentException e) {
        .....
        return false;
    }

    // Attente des données.
    long then = System.currentTimeMillis();
    long waitingPeriod = 16000; // timeout de 16 secondes
    try{
        synchronized (dataSync) {
            while (!dataReceived &&
                    System.currentTimeMillis() - then < waitingPeriod) {
                if (!dataReceived)
                    .....
                dataSync.wait(1000);
            }
        }
    } catch (Exception e) {}
    if (!dataReceived) {
        close();
        return false;
    }
    return true;
}

public boolean isDone() {
    return playerWindows.size() == 0;
}

// Fermeture des Players et des sessions
protected void close() {
    for (int i = 0; i < playerWindows.size(); i++) {
        try {
            ((PlayerWindow)playerWindows.elementAt(i)).close();
        } catch (Exception e) {}
    }
}

```

```

    }

playerWindows.removeAllElements();

// Fermeture session

for (int i = 0; i < mgrs.length; i++) {

    if (mgrs[i] != null) {

        mgrs[i].removeTargets( "Fermeture de la session à partir de AVReceive2");

        mgrs[i].dispose();

        mgrs[i] = null;

    }

}

} // fin protected void close()

PlayerWindow find(Player p) {

    for (int i = 0; i < playerWindows.size(); i++) {

        PlayerWindow pw = (PlayerWindow)playerWindows.elementAt(i);

        if (pw.player == p)

            return pw;

    }

    return null;

}

PlayerWindow find(ReceiveStream strm) {

    for (int i = 0; i < playerWindows.size(); i++) {

        PlayerWindow pw = (PlayerWindow)playerWindows.elementAt(i);

        if (pw.stream == strm)

            return pw;

    }

    return null;

}

/**/

* SessionListener.

*/



public synchronized void update(SessionEvent evt) {

    if (evt instanceof NewParticipantEvent) {

        ...

    }

}

```

```

}

/**
 * ReceiveStreamListener
 */
public synchronized void update( ReceiveStreamEvent evt) {
    RTPManager mgr = (RTPManager)evt.getSource();
    Participant participant = evt.getParticipant();           // Ne peut pas être Null.
    ReceiveStream stream = evt.getReceiveStream(); // Ne peut pas être Null.
    if (evt instanceof RemotePayloadChangeEvent) {

        ...
    }
    else if (evt instanceof NewReceiveStreamEvent) {
        try {
            stream = ((NewReceiveStreamEvent)evt).getReceiveStream();
            DataSource ds = stream.getDataSource();
            // Reconnaissance des formats.
            RTPControl ctl = (RTPControl)ds.getControl("javax.media.rtp.RTPControl");
            if (ctl != null){
                System.err.println(" - Reception de nouveaux flux RTP: " +
ctl.getFormat());
            } else
                System.err.println(" - Reception de nouveaux flux RTP");
            if (participant == null)
                //Aucun participant
            else {
                //Identification et retour chaîne= participant.getCNAMES().
            }
            // Cr éation d'un objet Player en passant Datasource comme argument à Media
            //Manager
            Player p = javax.media.Manager.createPlayer(ds);
            if (p == null)
                return;
            p.addControllerListener(this);
        }
    }
}

```

```

    p.realize();
    PlayerWindow pw = new PlayerWindow(p, stream);
    playerWindows.addElement(pw);
    // Renvoi du statut intialize() à la reception d'un flux
    synchronized (dataSync) {
        dataReceived = true;
        dataSync.notifyAll();
    }
} catch (Exception e) {
    //Gestion des exceptions
    return;
}
}

else if (evt instanceof StreamMappedEvent) {
    if (stream != null && stream.getDataSource() != null) {
        DataSource ds = stream.getDataSource();
        // Find out the formats.
        RTPControl ctl = (RTPControl)ds.getControl("javax.media.rtp.RTPControl");
        //Identification de l'envoyeur
        if (ctl != null)
            //contrôle format
            ...
    }
}

else if (evt instanceof ByeEvent) {
    System.err.println(" - \"bye\" de: " + participant.getCNAMES());
    PlayerWindow pw = find(stream);
    if (pw != null) {
        pw.close();
        playerWindows.removeElement(pw);
    }
}

}

}// fin AVReceive2

```

## La transmission de l'Audio et de la Vidéo en utilisant RTP

```
import javax.media.*;
import javax.media.rtp.*;
import javax.media.format.*;
import javax.media.control.*;
import javax.media.protocol.*;
public class StreamingServer {
    public static void main(String[] args)
    {
        //Chemin du fichier
        String FichierAdresse = "file:///C://lova//Test.avi";
        //Création du MediaLocator à partir du fichier
        .....
        //Déclaration du processeur
        .....
        try{
            //Création du Processor à partir du MediaLocator
            .....
            //Appel des fonctions qui vont permettre le lancement du
flux RTP
            configure(FichierCessor);
            SetSupportedFormat(FichierCessor);
            //passer dans l'état realized du processor
            realize(FichierCessor);
            //lancement
            Demarre(FichierCessor);
            launchRTPManager(FichierCessor);
        }
        catch(IOException e)
        {
            ...
        }
        catch(NoProcessorException e)
        {
            ...
        }
    }
}
```

```

public static Processor configure(Processor p)
{
    //Attendre tant que le Processor n'est pas configuré.
    // configuration du Processor
    return p;
}

public static void SetSupportedFormat(Processor p)
{
    //On met la description du contenu de sortie à RAW_RTP
    //On obtient les différentes pistes du processor
    TrackControl track[] = p.getTrackControls();
    for(int i = 0 ; i < track.length ; i++)
    {
        //on obtient les formats supportés pour cette piste
        Format suppFormats[] = track[i].getSupportedFormats();
        //Si il y a au moins un format supporté
    }
}

public static Processor realize(Processor p)
{
    //Attendre tant que le Processor n'est pas réalisé.
    while(p.getState() < Processor.Realized)
    {
        //Réalisation du Processor
        p.realize();
    }
    return p;
}

public static void Demarre(Processor p)
{
    //Demarrage du Processor
    p.start();
    System.err.println("started");
}

public static void launchRTPManager(Processor p)
{
    //Creation du DataSource correspondant au Processor
    //Nouvelle Instance d'un RTPManager
}

```

```
for(int i=0; i < pbss.length;i++)
{
    try{
        //Création d'une SessionAddress
        SessionAddress localaddr = new SessionAddress
        (InetAddress.getLocalHost(),port);
        //Initialisation du RTPManager
        //Création d'une SessionAddress
        //Ajout de cette SessionAddress dans le RTPManager
        //Creation d'un SendStream à partir du DataSource
        //Demarrage du SendStream
    }
    catch (Exception e)
    {
        //Gestion des exceptions
    }
}
}
```

## **ANNEXE 5 : QUELQUES DEFINITIONS**

### **Le codec**

Le mot codec vient de la contraction des 2 mots codeur et décodeur.

Un codec permet au programme qui l'utilise de comprendre un format pour lequel il n'est pas prévu à la base.

Exemple : un codec MP3 permet au programme qui l'utilise de sauvegarder et de lire les fichiers MP3.

On trouve ainsi des codecs vidéo et audio pour une multitude de programmes (et pour chaque version de ces programmes) Des fichiers, portant la même extension, peuvent toutefois être codés différemment.

### **L'encodeur**

Un encodeur est un petit programme réalisant l'encodage d'un fichier audio ou vidéo.

### **L'encodage**

L'encodage c'est la conversion d'un format en un autre.

### **Le concept de ports**

Lorsqu'un client contacte un serveur, c'est le plus souvent en vue d'utiliser un service précis (exemple : courrier électronique, web, FTP,...). Afin de différencier ces services, TCP dispose du concept de port qui permet à une même interface réseau de fournir plusieurs services différents.

### **Les numéros de port**

Les numéros de port sont des valeurs à 16 bits qui identifient les processus d'application ou des services réseaux. Dans chaque premier mot de l'en-tête de chaque datagramme TCP et chaque datagramme UDP, il y a le numéro de port source et les numéros de port destination. Le numéro de port source identifie le processus ayant envoyé les données. Le numéro de port destination identifie les destinataires.

C'est la combinaison de port et de protocole qui identifie de manière unique le processus spécifique auquel les données doivent être délivrées.

Les numéros de port inférieurs à 256 sont réservés aux services connus comme FTP, Telnet, HTTP. Les ports numérotés de 256 à 1024 sont utilisés pour des services spécifiques à UNIX, comme rlogin, etc.

## Les adresses IP

Sous la norme IPv4, une adresse IP est codée sur 4 octets. Une adresse IP identifie un réseau ou une machine.

Il y a ce qu'on appelle les classes d'adresses IP, résumées dans le tableau suivant :

Classe	Adresse IP
A	1.0.0.1 à 126.255.255.254
B	128.0.0.1 à 191.255.255.254
C	192.0.0.1 à 223.255.255.254
D	224.0.0.1 à 239.255.255.255
E	240.0.0.0 à 255.255.255.255

*Annexe 5.01 : les classes d'adresses IP.*

Les adresses IP privées, c'est-à-dire non routables sur Internet sont :

Classe	Adresse IP
A	10.0.0.0 à 10.255.255.255
B	172.16.0.0 à 172.31.255.255
C	192.168.0.0 à 192.168.255.255

*Annexe 5.02 : les adresses IP privées.*

Sous la norme IPv6, une adresse IP est codée sur 16 octets.

## Les numéros de protocole

Le numéro de protocole est un octet unique dans le troisième mot de l'en-tête du datagramme. La valeur identifie le protocole de la couche au-dessus de IP à laquelle les données doivent être passées.

. Voici quelques numéros de protocole :

Nom officiel du protocole	Numéro du protocole
---------------------------	---------------------

<b>IP</b>	0
<b>ICMP</b>	1
<b>IGMP</b>	2
<b>GGP</b>	3
<b>TCP</b>	6
<b>PUP</b>	12
<b>UDP</b>	17

*Annexe 5.03 : quelques exemples de numéros de protocoles.*

## BIBLIOGRAPHIE

- [1] C. Nowakowski et A.Roux, *Histoire des systèmes de télécommunication*, Dunod : Paris, 1994.
- [2] R.Besson, *Cours de télévision moderne*, Radio : Paris, 1988.
- [3] J.E. Boto Andrianandrasana, *Emetteur-récepteur-télévision*, Cours 4<sup>ème</sup> année, Dép. Tél.– E.S.P.A., A.U. : 2004-2005.
- [4] J.E. Boto Andrianandrasana, *Réseaux de vidéocommunication*, Cours 5<sup>ème</sup> année, Dép. Tél. – E.S.P.A, A.U. : 2005-2006.
- [5] [http://www.empreinte.com/HTML/webtv\\_architecture.htm](http://www.empreinte.com/HTML/webtv_architecture.htm).
- [6] P.Gasser, *Plate-forme Arts, Sciences, Technologies*, MSH, Paris, 2006.
- [7] [http://www-igm.univ-mlv.fr/\\_DR/XPOSE2005/BOITEL/streaming.html](http://www-igm.univ-mlv.fr/_DR/XPOSE2005/BOITEL/streaming.html).
- [8] A. Ratsimbazafy, *Téléinformatique et télématique*, Cours 4<sup>ème</sup> Année, Dép. Tél.-ESPA, A.U. : 2004-2005.
- [9] L.E. Randriarijaona, *TCP/IP*, Cours 4<sup>ème</sup> Année, Dép. Tél.-ESPA, A.U. : 2004-2005.
- [10] A.Pelat, *Le système de télévision*, Marketing : Paris, 1995.
- [11] UIT, *Recommandation : Rec UIT-R BT.470-4*, Düsseldorf, 1990.
- [12] C. Ratsihorarana, *Transmissions numériques*, Cours 5<sup>ème</sup> Année, Dép. Tél.-ESPA, A.U. : 2005-2006.
- [13] <http://drogo.cselt.stet.it/mpeg/>

- [14] G.Pujolle, *Les Réseaux*, Eyrolles : 2003.
- [15] A.Martin et A.Pelat, *Télévision numérique, compression et transmission du signal*, Ellipses : Paris, 1997.
- [16] H.Benoît, *Télévision par satellite : technique de la réception analogique et numérique*, Dunod : Paris, 1998.
- [17] <http://www.realnetworks.com/>
- [18] <http://www.streamingmedia.com/>
- [19] <http://dictionnaire.phpmyvisites.net/definition-ENCODEUR-4471.htm>
- [20] [http:// dictionnaire.phpmyvisites.net/Codec-audio&video-1.htm](http://dictionnaire.phpmyvisites.net/Codec-audio&video-1.htm)
- [21] [http://www.yahoo.com/Computers\\_and\\_Internet/Multimedia/Video/MPEG/](http://www.yahoo.com/Computers_and_Internet/Multimedia/Video/MPEG/)
- [22] <http://drogo.cselt.stet.it/mpeg/>
- [23] <http://www-ensimag.imag.fr/profs/cours/Exposes.Reseaux/Compression/mpegnav.html>
- [24] <http://www.bok.net/~tristan/MPEG/MPEG-content.html>

## PAGE DE RENSEIGNEMENTS

Nom : ANDRIANASOLO  
Prénoms : Lova Nirina  
Adresse de l'auteur : LOT VT 86 E Andohanimandrozeza  
ANTANANARIVO 101  
Tél : 033 14 182 97  
E-mail : louaimejava@yahoo.fr

Titre : **LA WEB TV**  
Rubrique :  
Nombre de pages : 92  
Nombre de tableaux : 09  
Nombre de figures : 37  
  
Mots clés : -Télévision  
-Internet  
-Streaming  
-WEB TV  
-Multicast  
-Unicast  
-JAVA  
-temps réel  
-RTP  
-UDP  
-RTCP  
-RTSP

Directeur de mémoire: Monsieur RATSIMBAZAFY Andriamanga

## **RESUME**

Cet ouvrage traite l'étude théorique de la Télévision sur Internet ou sur tout autre réseau IP. Il décrit les traitements effectués depuis l'acquisition du signal de télévision, sa compression, l'encodage, jusqu'à sa transmission sur Internet, même si les infrastructures réseaux existantes ne sont pas assez satisfaisantes pour offrir une qualité raisonnable.

Enfin, le développement de WebTVServer 1.0 et de WebTVPlayer 1.0 par nos propres soins permet d'exploiter au maximum le monde du Streaming, et de défier l'idée de transmettre tous les types de média, à travers tous les réseaux IP hauts débits, et ce, en temps réel.

## **ABSTRACT**

This work treats the theoretical survey of the Television on Internet or on all other IP network. It describes treatments done since the acquirement of the television signal, its compression, the encoding, until its transmission on Internet, even though infrastructures networks exist are not satisfactory enough to offer a reasonable quality.

Finally, the development of WebTVServer 1.0 and WebTVPlayer 1.0 by our own cares permits to exploit to the maximum hulls it of the Streaming, and to challenge the idea to transmit all types of media, through all IP's networks high debits, and that, in real time.